

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE

UNIVERSITE MENTOURI CONSTANTINE
FACULTE DES SCIENCES DE L'INGENIEUR
DEPARTEMENT D'ELECTRONIQUE

N° d'ordre :
Série :

Mémoire de Magistère

Présenté Par
AL ANI INES

Option
Contrôle

Thème

Commande supervisée d'un système à événements discrets Une approche par l'outil Grafcet

Membres du jury

Président	M. F Soltani	Professeur	Université Mentouri Constantine
Rapporteur	Mlle. N Mansouri	Professeur	Université Mentouri Constantine
Examineur	M. A Zatri	Professeur	Université Mentouri Constantine
Examineur	M. A Chaoui	Maître de conférences	Université Mentouri Constantine
Invité	M. H Hamdi	Chargé de cours	Université Mentouri Constantine

Année 2009

Remerciements

Je remercie tout d'abord Dieu Tout Puissant de m'avoir permis de terminer ce travail.

Je tiens à exprimer mes sincères remerciements à mon encadreur, Mr H.Hamdi, pour son soutien et ses précieux conseils.

Je remercie également Mr H.ALLA, inspirateur de ce sujet, pour ses orientations, sa documentation et ses judicieuses recommandations.

Je remercie Mlle N.Mansouri, mon encadreur administratif, qui m'a soutenue, conseillée et permis de mener ce travail à son terme.

Je tiens à exprimer ma reconnaissance et ma gratitude à tous mes professeurs, particulièrement ceux qui m'ont soutenue durant mes années d'études.

Enfin, je remercie les membres du jury pour avoir accepté de juger ce modeste travail.

AL-ANI INES

Dédicaces

Je dédie ce modeste travail à

Mes très chers parents qui m'ont éclairée le chemin du succès en me donnant la main tout au long de mes années d'études. C'est grâce à eux que j'en suis arrivée à ce stade.

Mes frères et sœurs qui ont toujours été près de moi pour m'encourager et me soutenir.

Ma très chère tante et sa famille.

Mon mari qui ma soutenu moralement et physiquement dans les plus difficiles moments.

Ma belle famille.

Ma famille en IRAQ.

L'âme de mes grands parents.

Mes amis ainsi que tous ceux qui ont été à mes cotés pour m'aider et me soutenir durant toutes mes années d'études.

INES

SOMMAIRE

<i>INTRODUCTION</i>	1
---------------------------	---

CHAPITRE I Supervision des systèmes à événements discrets

I.1. Principe de supervision	4
I.2. Techniques de base utilisées en supervision	5
I.2.a. Méthode basée sur les automates	5
I.2.b. Méthode de l'invariant	6
I.2.c. Méthode de sur état	8
I.2.d. Méthode structurelle	10

CHAPITRE II Supervision des SED modélisés par Grafcet

Introduction	12
II.1. Supervision par le principe de précedence	
II.1.a. Concepts de transitions contrôlable, incontrôlable, interdit	12
II.1.b. Concepts d'états admissible, frontière, admissible	14
II.1.c. Synthèse du superviseur	14
II.2. Application de la méthode de précedence sur le procédé SPT	22
II.2.a. Graphes d'états du SPT	23
II.2.b. Synthèse du superviser du procédé (SPT)	24
II.2.c. Fonctionnement du procédé SPT supervisé	30
II.3. Application de la méthode structurelle sur le procédé SPT	32
II.3.a. Réduction des contraintes par la méthode de l'invariant	33
II.3.b. Détermination des étapes de contrôle	34
II.3.c. Etablissement du Grafcet supervisé	37
II.4. Comparaison des résultats obtenus par les deux méthodes	37
II.5 Conclusion	38

CHAPITRE III Exemple d'implémentation sur API

Introduction	39
III.1. Description et principe de fonctionnement du procédé AME	39
III.2. Synthèse du superviseur	
III.2.a. Recherche d'états interdits	41
III.2.b. Détermination du superviseur	44
III.2.c. Fonctionnement du procédé supervisé	47
III.3. Implémentation du Grafset de commande supervisée	
III.3.a. Adressage des entrées / sorties.....	50
III.3.b. Programme séquentiel en langage PL7-2.....	51
III.3.c. Programme postérieur (actions) en langage LADDER.....	53
III.3.d. Programme préliminaire	54
III.4. Conclusion.....	56

CONCLUSION	57
-------------------------	-----------

Références bibliographiques

Liste des figures

Figure 1.1 Schéma modulaire du procédé étendu avec superviseur.....	4
Figure 2.1 Graphe d'Etats Accessibles	13
Figure 2.2 Détermination des étapes $E_{Pré}$ et E_{Suc}	19
Figure 2.3 Conditions de supervision vérifiées sur le GEA	20
Figure 2.4 Système de Pesage et Transport (SPT).	22
Figure 2.5 Grafquets de commande du SPT.....	23
Figure 2.6 Grafquet du procédé (SPT) non supervisé.	23
Figure 2.7 Graphe d'états accessibles du procédé (SPT) non supervisé.....	24
Figure 2.8 Graphe d'états accessibles simplifié du procédé (SPT) non supervisé.....	27
Figure 2.9 $E_{Pré}$ et E_{Suc} de l'étape E_3 sur le GEAS.....	27
Figure 2.10 $E_{Pré}$ et E_{Suc} de l'étape E_{12} sur le GEAS.....	28
Figure 2.11 Conditions de supervision sur le GEA du SPT.....	28
Figure 2.12 Graphe d'Etats Accessibles Simplifié du SPT.....	29
Figure 2.13 Grafquet de commande du procédé (SPT) supervisé.....	30
Figure 2.14 Graphe d'états accessibles sous superviseur du SPT.....	30
Figure 2.15 Grafquet de commande simplifié du procédé (SPT) supervisé.....	31
Figure 2.16 Couplage de la synchronisation avec la commande.....	32
Figure 2.17 Grafquet de commande du procédé avec synchronisation.....	32
Figure 2.18 Grafquet de commande du SPT avec le superviseur de Kattan.....	37
Figure 3.1 Identification des actionneurs.....	39
Figure 3.2 Identification des détecteurs de fin de course.....	40
Figure 3.3 Cycle de fonctionnement du procédé AME.....	40
Figure 3.4 Grafquets partiels du procédé AME.....	41
Figure 3.5 Grafquet de commande global de AME	42

Figure 3.6 Graphe d'état accessible du procédé AME non supervisé.....	43
Figure 3.7 Graphe d'états accessibles simplifié du procédé AME non supervisé.....	45
Figure 3.8 Le GEAS avec ordre d'occurrence de $E_{pré}$ et E_{suc}	46
Figure 3.9 GEA avec les conditions de précedence	47
Figure 3.10 Grafcet de commande supervisé de AME.....	48
Figure 3.11 Graphe d'états accessibles sous superviseur de AME.....	48
Figure 3.12 Grafcet de commande simplifié du procédé AME supervisé.....	49

Liste des tableaux

Tableau 2.1 Table de réduction sur états – contraintes.....	26
Tableau 3.1 Table de réduction sur états - contraintes.....	44
Table 3.2 Adresses des entrées.....	50
Table 3.3 Adresses des sorties.....	50

INTRODUCTION

INTRODUCTION

Dès ses origines, la *théorie des systèmes* et leur *commande* s'est intéressé à des systèmes physiques, décrits généralement par les *équations différentielles* ou aux *dérivées partielles*, auxquelles obéissent les phénomènes physiques correspondants.

En 1980 le monde de l'Automatique a pris en compte la gestion et la commande des "*Systèmes à Événements Discrets*" *SED*, où *l'espace d'état est un ensemble discret* et le changement d'état s'effectue à des instants précis. A chacune de ces transitions on peut associer un *événement*. Les variables d'état peuvent prendre des valeurs continues, par contre leur début et leur fin sont conditionnés par des événements.

Dans ces systèmes, l'essentiel de l'enchaînement dynamique des tâches provient du phénomène de *synchronisation*, nécessaire dans le cas de *l'exclusion mutuelle* ou de la *compétition* pour l'utilisation de ressources communes [1].

De nouveaux outils et de nouvelles méthodologies ont été développés afin de modéliser, analyser les performances et concevoir la commande de tels systèmes. Parmi les plus courants on peut citer les réseaux de files d'attente, le suivi de "trajectoire" des transitions [2], les réseaux de *Pétri* et algèbre linéaire [3], les automates et langage formel [4], la théorie des régions [5].

La modélisation des systèmes à événement discret utilise la notion de représentation modulaire. Chaque système est composé de deux parties, le *procédé étendu* et le *superviseur*.

Un superviseur est un *SED* qui permet de modifier le fonctionnement d'un procédé en interdisant la génération par le procédé de certains événements. Les techniques qui permettent de le synthétiser de façon systématique sont basées sur le concept de contrôlabilité, qui est la clef de la théorie de *Ramadge* et *Wonham* [6].

La théorie de la supervision des systèmes à événements discrets fournit des techniques qui permettent de garantir, à priori, que le fonctionnement du système supervisé respectera un cahier de charges imposé. Cette théorie, initiée par les travaux de *Ramadge* et *Wonham* dans le début des années 80, est basée sur l'utilisation des modèles automates et des langages formels. L'inconvénient de cette technique est l'explosion combinatoire du nombre d'états.

Charbonnier [7] a pu faciliter la synthèse du superviseur en combinant *Grafcet*, automates et théorie des langages. Il a ainsi démontré que dans le cas où le langage

des spécifications du superviseur est contrôlable par rapport au langage du procédé étendu, le *Grafcet* de spécification lui-même constitue le *Grafcet* du superviseur. Cependant le problème n'est pas résolu dans le cas où le langage des spécifications est non contrôlable.

Pour résoudre ce problème, plusieurs méthodes de synthèse de commande basées sur les Réseaux De Pétri "*RDP*" ont été proposées. Dans ce contexte l'ensemble des marquages interdits correspond aux séquences d'événements interdits. *Yamalidou* [8] a représenté les spécifications de fonctionnement traduisant les états qui doivent être interdits (marquages interdits) sous forme de contraintes. Son principal apport, qui a été capital pour les travaux qui ont suivi, a été la formulation sous forme mathématique des contraintes et de la matrice d'incidence du procédé. Chaque état interdit amène à une contrainte de fonctionnement.

Moody [9] a modifié les contraintes afin de construire les places de contrôle du superviseur optimal.

Mais le nombre de contraintes peut être important. *Dideban* et *Alla* [10] ont présenté des méthodes systématiques qui permettent de réduire le nombre de contraintes pour les réseaux saufs et conservatifs. *Dideban* a également proposé [11] une généralisation de la méthode pour l'étendre à des *RDP* non saufs.

Cependant toutes ces techniques présentent l'inconvénient majeur d'obtenir une commande qui n'est pas directement opérationnelle. En effet il faut passer par la phase de traduction sous forme de *Grafcet* (avec tous les inconvénients que cela comporte), pour pouvoir implémenter le résultat sur un Automate Programmable Industriel (*API*). *Kattan* [12] a appliqué la méthode de l'*invariant* pour synthétiser une supervision de commande basée sur le *Grafcet*. Cependant au moment de la création du superviseur, il réutilise le concept de matrice d'incidence du *RDP*. Au final les résultats obtenus ne diffèrent pas beaucoup de ceux obtenus avec les *RDP*. En plus d'être une approche structurée, la différence principale de la méthode est l'essai d'obtention du superviseur directement sous la forme d'un *Grafcet*.

Notre travail consiste à développer une technique qui permet de synthétiser un superviseur de commande qui soit optimal (le plus simple possible), en utilisant l'outil *Grafcet* de bout en bout. Le *Grafcet* sera utilisé dans toutes les phases de conception, de la modélisation du procédé et de sa commande à la synthèse du superviseur. Le gros avantage de l'utilisation du *Grafcet* est d'obtenir un résultat qui est directement implantable sur un *API*.

Ce document qui résume notre travail est structuré en trois chapitres. Le premier résume les travaux qui représentent une extension importante de la théorie de *Ramadge & Wonham*, et qui sont la base de la supervision des SED. Parmi ceux là, nous insisterons sur les concepts d'*invariant* et de *sur état*.

Le deuxième chapitre est consacré à l'exposé de notre technique. Après une brève définition des concepts de *transition contrôlable, incontrôlable et interdite*, on aborde les concepts d'*états acceptable, frontière et interdit*, déduits du graphe d'états accessibles. Ces concepts sont utilisés pour déterminer l'ensemble de contraintes équivalent au graphe, puis cet ensemble est réduit en utilisant la notion de *sur état*, pour aboutir à un ensemble d'étapes de contrôle. Cet ensemble est enfin minimisé grâce à la technique de *précédence* pour obtenir le superviseur optimal. A chaque étape de notre approche, nous gardons présent à l'esprit l'optimalité du modèle. Ensuite les résultats obtenus seront appliqués sur un exemple simple, dans un but de comparaison avec ceux obtenus par la technique de *Kattan* (méthode structurelle de supervision basée sur le *Grafset*).

Le troisième chapitre traite du volet implémentation. Nous appliquons la technique développée à l'exemple concret d'un poste d'alimentation, matriçage et évacuation de pièces. L'alimentation est effectuée par un robot électropneumatique, le matriçage et l'évacuation de ces pièces s'effectue par des vérins électropneumatiques. L'ensemble a un fonctionnement automatique, et nous traitons la conception et l'implémentation de la commande supervisée, sur l'automate programmable industriel TSX 17-20 en langage PL7-2.

CHAPITRE I

*Supervision des systèmes
à événements discrets*

I.1. Principe de supervision

Un système est constitué de trois modules : *procédé*, *commande*, *superviseur* (cf. figure 1.1).

Le procédé représente la partie physique du système avec tous les dispositifs de fonctionnement, qui seront pilotés par le module de commande. Le procédé fournira les événements concernant les fonctions qui doivent être accomplies pour que le module commande puisse donner les ordres d'action nécessaires au bon fonctionnement. On appelle procédé étendu l'ensemble constitué du procédé et de la commande.

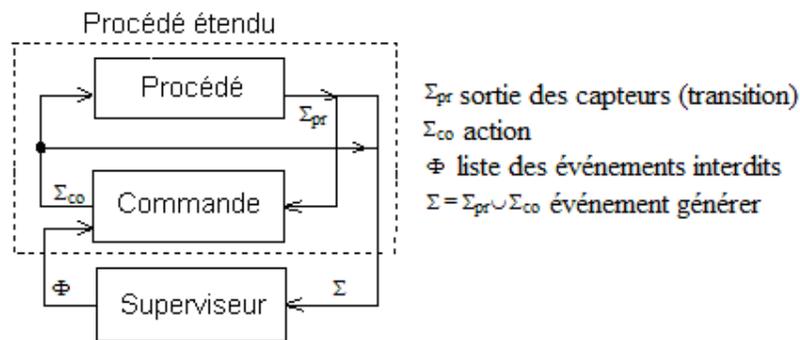


Figure 1.1 Schéma modulaire du procédé étendu avec superviseur.

Pour illustrer l'utilité du superviseur on doit faire appel à la notion de la contrôlabilité des SED.

La *contrôlabilité* est le concept le plus important sur lequel sont basées toutes les études établies dans ce domaine. Dans le procédé on a des actions et des événements, les actions sont les ordres Σ_{co} envoyés par la commande vers le procédé et les transitions Σ_{pr} représentent la réponse du procédé aux ordres reçus. La commande a le pouvoir de gérer les actions du procédé mais non pas celui de les interdire car elle ne fait pas la différence entre les événements contrôlables Σ_c et incontrôlables Σ_u ($\Sigma_{co} \in \Sigma_c \cup \Sigma_u$, $\Sigma_{pr} \in \Sigma_c \cup \Sigma_u$), ceci est le rôle du superviseur.

Un événement contrôlable est un événement qui peut être interdit ou autorisé par un superviseur, mais ce dernier ne peut effectuer la même opération avec les événements incontrôlables car ils ne le permettent pas.

L'occurrence des événements incontrôlables provoque des défaillances dans le fonctionnement du procédé. Le rôle du superviseur sera de prédire ces défaillances

puis les interdire en intervenant sur les événements contrôlables, pour assurer que le fonctionnement du système en boucle fermée (procédé bouclé avec le superviseur) réponde aux exigences de fonctionnement décrites dans le cahier de charges.

Pour synthétiser le superviseur soit on a recours aux automates & langages formels, soit on procède à la synthèse directe en se basant sur l'outil utilisé pour modéliser la commande, c'est le *RDP* ou le *Grafcet*.

Les modules de commande et de supervision sont séparés. Cela ajoute des performances à la modélisation, car si on modifie les spécifications ou contraintes de fonctionnement à imposer au niveau de la supervision, le système de commande ne doit pas être modifié. Seul le superviseur sera modifié. Il en est de même pour les systèmes modulaires constitués de sous systèmes, et par voie de conséquence de modules de commande propres à ces sous systèmes, ceci facilitera la modification et l'extension. Cette caractéristique est à la base de la modélisation des systèmes à événements discrets.

I.2. Techniques de base utilisées en supervision

I.2.a. Méthode basée sur les automates

Cette technique [6] présente les principes de base de la supervision, elle est basée sur les automates et les langages formels pour synthétiser le superviseur. En plus de l'explosion combinatoire du nombre d'états, l'application de cette technique est lourde car on effectue beaucoup de transformations pour aboutir au modèle automate du procédé sous supervision.

Pour résoudre ce problème d'autres techniques ont été développées. Parmi les plus importantes on peut citer celle de *charbonnier* [7] où la commande et la supervision sont modélisées par le Grafcet, mais le test de la contrôlabilité est implémenté sur le modèle automate.

La commande de chaque sous système est représentée par un Grafcet indépendant, le superviseur doit quant à lui assurer le respect des spécifications de commande.

Puis il faut modifier les Grafcets de commande en leur rajoutant des conditions pour qu'ils prennent en compte la supervision. Ces conditions doivent être reportées sur les événements contrôlables.

Le Grafcet de commande sous supervision de chaque sous système sera ensuite exprimé par un modèle automate, et le Grafcet de supervision sera exprimé par une machine de Moore. Puis Tous ces automates seront combinés pour obtenir le modèle automate de la commande sous supervision.

Ces transformations sont faites pour pouvoir exprimer le superviseur et la commande par un modèle accepteur, afin de vérifier la contrôlabilité des spécifications de supervision par rapport au langage du procédé étendu (commande), par le biais de l'algorithme de *Kumar* [13]. Si la contrôlabilité n'est pas satisfaite, il faut refaire la synthèse du modèle accepteur du langage suprême contrôlable (le plus grand langage contrôlable) du fonctionnement désiré.

On remarque que l'avantage de cette étude réside dans l'utilisation du Grafcet comme outil de modélisation, mais la synthèse du contrôleur est très lourde.

Le problème d'explosion combinatoire est résolu uniquement en partie, car il ne l'est pas dans le cas où le langage des spécifications est non contrôlable.

De plus la commande est modifiée (plus précisément ses Grafcets) selon les spécifications de supervision, ce qui ne répond pas au but de la modélisation en modules.

I.2.b. Méthode de l'invariant

Yamalidou [8] utilise le réseau de *Pétri* pour modéliser la commande et la supervision (*RDP* sauf), et le *quasi-RDP* pour synthétiser le superviseur de commande, le but étant d'obtenir un graphe de marquage qui satisfait les spécifications de fonctionnement en boucle fermée. L'occurrence d'événement incontrôlable provoque des défaillances dans le fonctionnement, ces défaillances sont représentées par des états appelés états interdits et frontières (un état correspond a un ensemble de places marquées), donc l'interdiction d'état revient à interdire le marquage équivalent.

Soit $P_1P_2\dots P_n$ un état interdit, alors le marquage $m_1m_2\dots m_n$ sera interdit.

Cela se traduit sous forme de contrainte linéaire par :

$$\sum_{i=1}^n m_i \leq n - 1 \quad (1.1)$$

qui indique que la somme des marquages m_i doit être inférieure ou égale au nombre de places n moins un, pour s'assurer qu'au moins une place est non marquée (un état non atteint).

Pour satisfaire cette condition on ajoute au *RDP* des places de contrôle, qui seront définies à partir des contraintes. Le nombre de places de contrôle dépend du nombre de contraintes.

Soit un procédé *P* dont la commande est modélisé par un *RDP* constitué de *p* places et *t* transitions, dont la matrice d'incidence est donnée par $Dp(p.t)$.

Le nombre de contraintes *k* (égal au nombre d'états frontières) sera exprimé sous la forme de : $\sum_{i=1}^n m_i \leq b$ (*n* est le nombre de places dans chaque contrainte, *b* le nombre max de marquages qui ne doit pas être dépassé).

Chaque contrainte permet de définir un vecteur L_j de dimension (1.*p*). Les places qui appartiennent à la contrainte prennent la valeur 1 dans L_j , et les autres prennent 0. Comme *j* varie de 1 jusqu'à *k* on peut alors construire la matrice des contraintes $L(k.p)$ depuis l'ensemble de contraintes :

$$L(k.p) = \begin{bmatrix} L_1(1.p) \\ \cdot \\ \cdot \\ L_k(1.p) \end{bmatrix} \quad (1.2)$$

$$\text{La matrice d'incidence } D_C \text{ du contrôleur est définie par : } D_C = -L.D_p \quad (1.3)$$

où chaque ligne de la matrice D_C représente une place de contrôle (D_C a une dimension (*k.t*)), les transitions qui possèdent 1 et -1, seront utilisées comme sortie et entrée pour la place de contrôle.

Le marquage initial des places de contrôle sera obtenu à partir de la formule $\mu_{c0} = b - L.\mu_{p0}$, (1.4a)

où μ_{c0} désigne le marquage initial des places de contrôle, et μ_{p0} celui des places du procédé. Si on exprime les marquages sous forme vectorielle, la relation 1.4 devient :

$$\begin{bmatrix} P_{c1} \\ \cdot \\ P_{ck} \end{bmatrix} = \begin{bmatrix} b_1 \\ \cdot \\ b_k \end{bmatrix} - \begin{bmatrix} L_1(1.p) \\ \cdot \\ L_k(1.p) \end{bmatrix} \bullet \begin{bmatrix} P_{p1} \\ \cdot \\ P_{pp} \end{bmatrix} \quad (1.4b)$$

La technique de *Yamalidou* permis de trouver les places de contrôle nécessaires à l'élimination des contraintes du fonctionnement. Son inconvénient est le nombre important de places de contrôle, car chaque contrainte introduit une place de contrôle. Pour réduire ce nombre *Yamalidou* a développé la méthode de l'*invariant*.

Le *quasi-RDP* exprime tous les états par lesquels passera le système (procédé étendu avec les spécifications de fonctionnement). Une transition incontrôlable Σ_U est franchie si les places situées en amont et appartenant au *RDP* de commande d'un procédé (ou sous système) sont marquées (elle n'inclue pas les spécifications de la commande), indépendamment du marquage des places situées en amont de cette transition, et appartenant aux *RDP* des autres sous systèmes.

Les transitions qui provoquent des défaillances dans le système sont appelées transitions interdites. Tout état placé en amont est un état interdit. S'il possède au moins une transition contrôlable en amont il sera traité comme état frontière. Sinon on cherche le premier état qui se situe en amont de l'état interdit et qui dispose au moins d'une transition contrôlable en amont pour qu'il soit un état frontière.

D'après les états frontières on constitue l'ensemble des contraintes linéaires. Pour réduire ce nombre on fait appel à **la méthode de l'invariant**.

Cette technique permet de remplacer r contraintes constituées de n places par une seule contrainte constituée de $n-1$ places, si certaines conditions sont vérifiées :

- Il faut que les r contraintes possèdent $n-1$ places communes,
- les places non communes exprimées sous forme de vecteur X satisfont l'équation de l'*invariant* $X^T(1.p) \cdot D_p(p.t) = 0$ (donc forment un ensemble de places X_i qui ne peuvent jamais être marquées en même temps, ***invariant***).

Soient r états frontière $P_1P_k...P_j$, $P_2P_k...P_j$, ..., $P_rP_k...P_j$. Chaque état est constitué de n places ($n > 2$), exprimées sous forme de contrainte linéaire par :

$$\left. \begin{array}{l} M_1 + M_k + \dots + M_j \leq n-1 \\ M_2 + M_k + \dots + M_j \leq n-1 \\ \dots \\ M_r + M_k + \dots + M_j \leq n-1 \end{array} \right\} \quad (1.5)$$

Si $M_1 + M_2 + \dots + M_r = 1$, les places P_1, \dots, P_r forment un *invariant*, donc on peut remplacer les r contraintes linéaires par une seule:

$$M_k + \dots + M_j \leq n-2 \quad (1.6)$$

Dans le cas où l'équation d'*invariant* n'est pas vérifiée cette technique de réduction de contraintes n'est plus valable.

I.2.c. Méthode de sur état

Le travail de *Dideban* [11] est complémentaire à celui de *Yamalidou*. Il a utilisé la même démarche pour trouver les places de contrôle et déterminer le marquage initial, mais pour réduire les contraintes il a développé la méthode de *sur état*.

Dideban considère que si l'ensemble des états interdits possède un élément commun P_i alors on peut empêcher l'occurrence de ces états en interdisant l'occurrence de cet élément P_i , appelé *sur état*.

L'approche revient à interdire un nombre réduit de *sur état*, mais qui couvre le maximum de contraintes, à condition qu'il n'inclue pas de *sur état* admissible.

La technique appliquée est résumé comme suit :

Pour chaque état frontière de dimension n , on réalise le *sur état* b_i qui représente l'ensemble des états obtenus par toutes les combinaisons possibles des places constituant la contrainte, et dont la dimension est inférieure ou égale à n .

Soit P_1P_3 un état frontière, son *sur état* est donné par $b_1 = P_1, P_3, P_1P_3$.

L'union des b_i forme l'ensemble global de *sur état* frontière $B_1 = \left\{ \bigcup_{i=1}^n b_i \right\}$.

De la même façon on constitue l'ensemble de *sur état* admissible $A_1 = \left\{ \bigcup_{i=1}^m a_i \right\}$.

On peut retrouver dans la liste B_1 soit des *sur états* admissibles (appartenant à l'ensemble A_1), soit redondants (couverts par un autre *sur état*). Ils seront supprimés pour obtenir l'ensemble final de *sur état* B_2 .

On constitue les contraintes linéaires depuis l'ensemble B_2 . Une deuxième simplification est possible grâce à la technique de fusionnement, où r contraintes peuvent être remplacées par une seule, à condition que:

- Les contraintes possèdent n places communes,
- les combinaisons deux à deux des éléments non communs ne font pas partie de l'ensemble de *sur état* admissibles.

Soit $B_2 = P_xP_{i1}, P_xP_{i2}, \dots, P_xP_{ir}$,

si $\{P_{i1}P_{i2}, P_{i1}P_{i3}, \dots, P_{i1}P_{ir}, P_{i2}P_{i3}, P_{i2}P_{i4}, \dots, P_{i2}P_{ir}, \dots, P_{ir-1}P_{ir}\} \notin A_1$

alors on peut écrire $M_{i1} + M_{i2} \dots + M_{ir} \leq 1$.

Donc les r contraintes ($M_{i1} + M_x \leq 1$, $M_{i2} + M_x \leq 1$, .., $M_{ir} + M_x \leq 1$) linéaires seront remplacées par une seule $M_x + M_{i1} \dots + M_{ir} \leq 1$.

L'ensemble réduit de contraintes obtenues depuis la technique de *sur état* et de fusionnement permet de faire le choix des éléments qui couvrent toutes les contraintes linéaires.

Réduire le nombre de contraintes entraîne la réduction du nombre de places de contrôle.

Cette combinaison de techniques (*Invariant* et *sur état*) a permis de faciliter la synthèse du superviseur de contrôle, pour un procédé dont la commande est effectuée par l'outil *RDP*.

L'inconvénient réside dans l'implémentation de la commande supervisée sur les Automates Programmable Industriel, qui n'est pas évidente. La solution qui a été appliquée consiste à transformer le *RDP* de la commande supervisée en Grafcet, car elle représente la voie d'implémentation directe.

1.2.d. Méthode structurelle

Pour résoudre le problème d'implémentation *Kattan* [12] a adapté une méthode, en utilisant le *Grafcet* comme outil de modélisation. Mais pour synthétiser le superviseur il a appliqué directement les résultats obtenus à partir des études basées sur la modélisation *RDP*.

Vu que le Grafcet de commande a une structure similaire à celle du *RDP* sain, *Kattan* a pu utiliser la technique de l'invariant pour synthétiser le superviseur exprimé sous forme matricielle, depuis la matrice du procédé et la matrice des contraintes. Il a eu un *superviseur* qui a des performances identiques à celui obtenu par l'utilisation du *RDP*.

Cette technique a permis de profiter des avantages apportés par l'outil de modélisation Grafcet dans l'implémentation sur *API*, donc de résoudre les difficultés d'implémentation causées par le *RDP*. De plus contrairement à celle de Charbonnier [7], elle ne modifie pas les Grafcets de commande des sous systèmes.

Cependant elle n'est pas optimisée, aussi bien pour le nombre d'étapes que celui des contraintes.

Notre but sera de profiter des avantages d'implémentation directe du Grafcet, en développant une technique basée sur le Grafcet de bout en bout : de la modélisation du procédé à l'implémentation sur API en passant par la synthèse du superviseur sous forme d'étapes de Grafcet. De plus le superviseur synthétisé devra être optimal.

On profitera bien entendu des résultats et des théorèmes obtenus pour le *RDP* sauf compte tenu de la nature discrète des événements générés par le Grafcet, et de la nature booléenne du franchissement des transitions.

Comme notre technique est également structurelle (basée sur la structure du Grafcet plutôt que le développement de grammaires et langages), la méthode de *Kattan* sera utilisée comme support de comparaison afin de déterminer les performances de notre technique.

CHAPITRE II

*Supervision des SED
modélisés par Grafcet*

Introduction

Les études basées sur le *RDP* pour commander les *systèmes à événements discrets* ont permis d'obtenir une technique qui facilite la synthèse du superviseur, qui est exprimée par des places de contrôle. Le nombre d'arcs n liés à chaque place de contrôle dépend du nombre p_i de places qui constituent la contrainte linéaire ($n=2^{p_i}$). Une contrainte est constituée au minimum de deux places donc le nombre d'arcs sera au minimum 4.

Kattan [12] a utilisé les résultats développés pour les études de synthèse basées sur le *RDP* en les appliquant à la synthèse basée sur le *Grafcet*. Il a adopté le même principe de synchronisation que celui du *RDP*, ainsi que la technique de réduction du nombre de contraintes.

Notre objectif est d'utiliser le *Grafcet* de bout en bout (de la spécification jusqu'à l'implémentation sur *API*) tout en nous efforçant de synthétiser le superviseur, le plus simple possible, qui satisfait les spécifications de fonctionnement.

II.1. Supervision par le principe de précédence

On va utiliser le *Grafcet* pour commander et synchroniser le fonctionnement du procédé, les contraintes imposées par les transitions incontrôlables seront réduites par la méthode du *sur état*. Pour aboutir au plus simple superviseur de contrôle on a développé une méthode basée sur le principe de "*précedence*" qu'on a appelée "*technique de prédécesseur et de successeur*".

Dans ce qui suit, on décrit le détail des étapes à suivre pour obtenir le *superviseur*.

II.1.a. Concepts de transitions contrôlable, incontrôlable, interdit

On réalise les *Grafcets* de commande (pour chaque sous système) ainsi que le *Grafcet* de spécification qui permet la synchronisation entre ces derniers. Ensuite on relie tous ces *Grafcets* pour obtenir le *Grafcet* de commande global qui permet d'obtenir le graphe "d'états accessibles" *GEA*.

Le *GEA* décrit le comportement du *Grafcet* de commande avec synchronisation. Le changement d'une étape X_i du *Grafcet* (activation ou désactivation) provoque un changement d'état de E_i vers E_j du *GEA*. Un état du graphe sera représenté par une bulle qui porte les numéros des étapes actives du *Grafcet* (cf.figure 2.1).

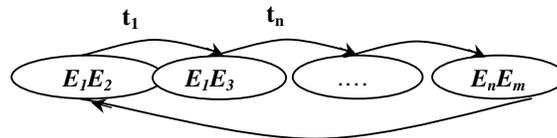


Figure 2.1 Graphe d'Etats Accessibles.

Le passage d'un état accessible E_i vers un autre E_j se fait par le franchissement d'une transition t_i du Grafcet. On attribue à l'arc qui relie les deux états accessibles, l'information sur l'identité de la transition franchie. On a trois représentations différentes de transitions:

- \rightarrow Transition contrôlable ou à événement contrôlable.
- \rightarrow Transition incontrôlable ou à événement incontrôlable.
- $--\rightarrow$ Transition interdite ou à événement interdit.

Définition 1: Transition contrôlable

Les transitions sont *contrôlables* (en d'autres termes les événements associés sont contrôlables), quand le franchissement de ces transitions, si elles sont validées par l'activité de leur(s) étape(s) amont, peut être autorisé ou interdit par une action du contrôleur autorisant ou pas la prise en compte de l'occurrence de l'événement qui leur est associé.

Définition 2: Transition incontrôlable

Les transitions sont dites *incontrôlables* (en d'autres termes les événements associés sont non contrôlables), quand le franchissement de ces transitions, si elles sont validées par l'activité de leur(s) étape(s) amont, sur occurrence de l'événement qui leur est associé, ne peut en aucun cas être interdit.

Un événement incontrôlable (correspondant au franchissement d'une transition incontrôlable) peut poser problème au niveau de la synchronisation, correspondant à une convergence en *ET* représentée par une sortie de séquence simultanée. Si le sous système force le passage de la transition sans attendre l'information de synchronisation apportée par la spécification, on aboutit à une situation interdite qui ne satisfait pas les spécifications de fonctionnement.

Donc implicitement ce type de transition va provoquer des problèmes de synchronisation.

Définition 3 : Transition interdite

On dit qu'une *transition* (à événement incontrôlable) est *interdite* s'il s'agit d'une transition aboutissant à une situation interdite, qui ne satisfait pas les spécifications de fonctionnement.

II.1.b. Concepts d'états admissible, frontière, interdit

Le superviseur doit assurer l'accomplissement des exigences de fonctionnement. Au préalable il faut d'abord extraire tout type d'anomalie rencontrée dans le graphe d'états accessibles. Dans ce contexte on définit trois types d'états :

Définition 4 : Un état interdit est un état qui ne doit jamais être atteint car il a en aval au moins une transition interdite. L'ensemble des états interdits est représenté par E_I .

Pour pouvoir exclure ces états ils doivent avoir au minimum une transition contrôlable en amont, mais ce n'est pas le cas dans la majorité des situations, donc on fait appel aux **états frontières**.

Définition 5 : état frontière est un état interdit qui dispose au minimum d'une transition contrôlable en amont. Si cette condition n'est pas vérifiée l'état frontière sera l'état situé en amont de l'état interdit et qui dispose d'une transition contrôlable en amont. L'ensemble des états frontière est représenté par E_B .

Définition 6 : état admissible est un état qui n'est ni un état frontière ni un état interdit, c'est donc un état qui satisfait les spécifications de fonctionnement.

Exprimé autrement, on peut dire que c'est un état dont on est sûr que l'activation ne provoquera pas d'aléa de fonctionnement dans le système.

L'ensemble des états frontière est représenté par E_A .

II.1.c. Synthèse du superviseur

Pour empêcher le franchissement des transitions incontrôlables qui aboutissent à des situations interdites, on interdit l'accès aux états défendus (précédant les transitions "indésirables") depuis les états autorisés immédiatement en amont. Cela se traduit par l'interdiction du franchissement d'une transition contrôlable.

L'algorithme, basé sur l'interdiction des états frontières, comprend les étapes suivantes :

- détermination des contraintes par les états frontières,
- réduction du nombre de contraintes par la méthode du *sur état*,
- calcul du superviseur par la méthode de *précédence*,

- simplification et constitution du superviseur,
- vérification du fonctionnement du système en boucle fermée.

II.1.c.1. Transformation des états frontières en contraintes linéaires

Si $E_1E_2\dots E_n$ est un état frontière constitué de n étapes, l'interdire revient à dire qu'on doit avoir au moins une étape inactive. On peut exprimer cela sous forme de contrainte linéaire par
$$\sum_{i=1}^n E_i \leq n-1 \quad (2.1)$$

($E_i = 1$ correspond à l'activation de l'étape E_i du *Grafcet*).

Les états frontières seront tous exprimés sous la forme de contraintes linéaires. Une étape de contrôle est une étape ajoutée au *Grafcet* initial qui permet de satisfaire ces contraintes. Le nombre d'étapes de contrôle dépendra du nombre de contraintes. Ce dernier sera réduit par la méthode du *sur état*.

II.1.c.2. Réduction du nombre de contraintes par la méthode du sur état

Dans cette méthode au lieu d'interdire n états frontières constitués de x étapes, on interdira r états frontière constitués de y étapes ($r < n$, $y < x$), mais qui couvrent toutes les contraintes, à condition qu'il n'y ait pas d'état admissible inclus dans l'interdiction (n, r, x, y : naturels positifs).

Le nombre de contraintes ainsi que le nombre d'étapes présentées dans chaque contrainte sera réduit.

En premier lieu il faut construire les ensembles de *sur états* frontières et admissibles, puis ôter de l'ensemble des *sur états* frontières les éléments qui appartiennent à l'ensemble des *sur états* admissibles. Ensuite pour obtenir l'ensemble réduit des *sur états* frontières il suffit d'éliminer les états redondants.

a. Ensembles de sur états frontières et admissibles

Définition7 : sur état

Soit $E_i = \{ E_{i1}E_{i2}\dots E_{im} \}$ un état du système. L'ensemble des *sur états* de E_i noté $\mathcal{E}_i^{\text{sur}}$ est égal à l'ensemble des parties de E_i sans l'ensemble vide.

En d'autres termes le *sur état* $\mathcal{E}_i^{\text{sur}}$ est l'ensemble de toutes les combinaisons d'étapes possibles obtenues à partir de l'état E_i .

Ainsi par exemple pour l'état frontière $b_i = E_1E_2E_3$, l'ensemble de ses *sur états* est donné par $\mathcal{E}_i^{\text{sur}} = \{ E_1, E_2, E_3, E_1E_3, E_1E_2, E_2E_3, E_1E_2E_3 \}$.

L'union des $\mathcal{G}_i^{\text{sur}}$ forme l'ensemble \mathcal{B}_1 global des *sur états* frontières $\mathcal{B}_1 = \bigcup_{i=1}^n \mathcal{G}_i^{\text{sur}}$ ($n = \text{card}(E_B)$ est le nombre d'états frontière).

Il en est de même pour l'ensemble des états admissibles E_A qui donnera l'ensemble des *sur états* admissibles $\mathcal{A}_1 : \mathcal{A}_1 = \bigcup_{i=1}^m \mathcal{a}_i^{\text{sur}}$ ($m = \text{card}(E_A)$ est le nombre d'états admissibles)

b. Suppression des sur états admissibles de l'ensemble des sur états frontières

Par construction de l'ensemble des *sur états* frontières \mathcal{B}_1 , on peut avoir introduit des *sur états* admissibles appartenant à \mathcal{A}_1 . Il faut alors les supprimer.

$\mathcal{B}_2' = \mathcal{B}_1 \setminus (\mathcal{B}_1 \cap \mathcal{A}_1)$: on supprime de la liste \mathcal{B}_1 tous les éléments qui appartiennent à \mathcal{A}_1 .

c. Suppression des sur états non minimaux

Quand un état est déjà couvert par un autre *sur état* de plus petit cardinal, on dit qu'il n'est pas minimal. Pour éviter la redondance, on supprime de la liste \mathcal{B}_2' tous les *sur états* non minimaux.

$B_2 = B_2' - \{b_{2j} \in B_2' / \exists b_{2i} \in B_2' \wedge b_{2j} \subset b_{2i}\}$: il faut supprimer b_{2j} de la liste \mathcal{B}_2' si b_{2j} est un sous ensemble de b_{2i} qui est inclus dans \mathcal{B}_2' .

d. Réduction du nombre de sur états

Depuis la liste \mathcal{B}_2 on choisit les *sur états* qui couvrent toutes les contraintes linéaires. Pour le faire on construit d'abord un tableau de *sur états* et de contraintes : les *sur états* en lignes et les contraintes en colonne. Puis en adoptant une technique proche de celle de *Mc Cluskey* en logique combinatoire [14], on attribue à chaque *sur état* la liste des contraintes qui le couvrent. L'ensemble de contraintes réduit \mathcal{C}_2 est constitué des *sur états* qui couvrent le maximum de contraintes tout en éliminant la redondance.

L'algorithme de construction de l'ensemble \mathcal{C}_2 est le suivant :

- *Lecture des tableaux \mathcal{B}_2 (sur états) et E_B (contraintes).*
- *Construction du tableau D et D_{suite} initialement vide.*

- Prendre un élément de \mathcal{B}_2 et l'écrire dans D. Lui associer (c'est-à-dire écrire dans D_{suite}) tous les éléments de E_B couverts par cet élément.

- Refaire le même travail pour tous les éléments de \mathcal{B}_2 .

- *Construction du tableau C_2*

- Lire le premier élément de D et le mettre dans C_2 .

- Prendre l'élément suivant de D. Comparer ses composantes associées aux composantes des éléments présents dans C_2 :

*si une contrainte est déjà couverte (redondante) par un *sur état* précédent, on la supprime pour le *sur état* qui possède le minimum de contraintes (on la garde pour le *sur état* qui couvre le plus grand nombre de contraintes) ;

*puis vérifier

- si le *sur état* en cours possède zéro contraintes, il est éliminé,

- sinon il est enregistré dans C_2 .

- Refaire le même travail pour tous les éléments de D

- Supprimer du tableau C_2 tous les éléments qui ne possèdent pas de contraintes.

e. fusionnement des contraintes

Soient $\{E_i E_{i1}, E_i E_{i2}, E_i E_{i3}, \dots, E_i E_{in}\}$ des éléments de C_2 , où E_i , représente une étape commune (E_{CM}) à toutes les contraintes, et $E_{i1}, E_{i2}, E_{i3}, \dots, E_{in}$ représentent les étapes non communes (E_{N-CM}). Si aucun des éléments de l'ensemble $\{E_{i1} E_{i2}, E_{i1} E_{i3}, \dots, E_{i1} E_{in}, E_{i2} E_{i3}, E_{i2} E_{i4}, \dots, E_{i2} E_{in}, \dots, E_{in-1} E_{in}\}$, obtenu par combinaison des étapes non communes, n'appartient à l'ensemble des *sur états* admissibles \mathcal{A}_1 (cela signifie que les étapes non communes $E_{i1}, E_{i2}, E_{i3}, \dots, E_{in}$ ne sont jamais toutes actives en même temps), alors on peut fusionner les contraintes en écrivant [11]:

$$\left. \begin{array}{l} E_i + E_{i1} \leq 1 \\ E_i + E_{i2} \leq 1 \\ E_i + E_{i3} \leq 1 \\ \dots \\ E_i + E_{in} \leq 1 \end{array} \right\} \Leftrightarrow E_i + E_{i1} + E_{i2} + E_{i3} + \dots + E_{in} \leq 1 \quad (2.2)$$

$C_1 = \{ (E_i, E_{i1} E_{i2} E_{i3} E_{in}) \}$, ensemble de *sur états* après fusionnement.

Si $\{E_{i1}E_{i2}, E_{i1}E_{i3}, \dots, E_{i1}E_{in}, E_{i2}E_{i3}, E_{i2}E_{i4}, \dots, E_{i2}E_{in}, \dots, E_{in-1}E_{in}\} \in \mathcal{A}1$, cela signifie que la condition de fusionnement est non vérifiée, alors C_1 sera égal à C_2 .

II.1.c.3. Construction du superviseur de commande par le principe de précedence

a. Application du principe de successeur et prédécesseur

On doit construire le superviseur qui satisfait les conditions de l'ensemble C_1 . Les éléments de l'ensemble C_1 peuvent être fusionnés ou non:

- Si les éléments de C_1 sont non fusionnés et représentés par $E_x E_y \dots E_w$, ces étapes ne doivent pas être actives en même temps,

$$E_x E_y E_z \dots E_w \in C_1 \Rightarrow E_x + E_y + E_z + \dots + E_w \leq n - 1 \quad (2.3)$$

Il faut que la désactivation de chaque étape de la contrainte (notée $\downarrow E_i$) permette (\mapsto) l'activation (notée $\uparrow E_i$) des autres. Si le nombre d'étapes qui constituent la contrainte augmente, implicitement le nombre de conditions augmente.

$$\begin{aligned} E_x + E_y + E_z + \dots + E_w \leq n - 1 \Leftrightarrow & (\downarrow E_x \mapsto \uparrow E_y \cdot \uparrow E_z \cdot \dots \cdot \uparrow E_w) \wedge \\ & (\downarrow E_y \mapsto \uparrow E_x \cdot \uparrow E_z \cdot \dots \cdot \uparrow E_w) \wedge \\ & (\downarrow E_z \mapsto \uparrow E_x \cdot \uparrow E_y \cdot \dots \cdot \uparrow E_w) \wedge \\ & \dots \wedge \\ & (\downarrow E_w \mapsto \uparrow E_x \cdot \uparrow E_y \cdot \uparrow E_z \cdot \dots) \end{aligned} \quad (2.4)$$

Donc on aura besoin de n conditions pour éviter l'activation simultanée de ces étapes.

Chaque condition sera exprimée par une étape de contrôle. Le nombre d'arcs liés à cette étape est égal au nombre d'étapes qui constituent chaque condition: un arc entrant provenant de l'étape désactivée, et $n-1$ arcs sortants pour valider l'activation des $n-1$ étapes restantes.

Le but est de synthétiser un superviseur qui accomplit la même tâche mais qui a une taille réduite. Pour cela il faut réduire le nombre de conditions ainsi que le nombre d'étapes qui constituent chaque condition.

Notre approche consiste à dire qu'au lieu que la désactivation de chaque étape de la contrainte autorise l'activation des autres, il suffit qu'elle autorise l'activation de l'étape qui lui succède dans le "graphe d'états accessibles simplifié" *GEAS*¹ du procédé non supervisé. Ainsi l'ordre d'occurrence des étapes qui constituent la contrainte va définir les conditions d'activation.

Considérons l'exemple précédent et supposons que :

$E_x E_y E_z \dots E_w \in C_1 \Rightarrow E_x + E_y + E_z + \dots + E_w \leq n - 1$, où l'ordre d'occurrence dans le *GEAS* du procédé non supervisé est donnée par : $E_y, E_x, E_z, \dots, E_w$. Par conséquent

$$E_x + E_y + E_z + \dots + E_w \leq n - 1 \Leftrightarrow (\downarrow E_y \mapsto \uparrow E_x) \wedge (\downarrow E_x \mapsto \uparrow E_z) \wedge \dots \wedge (\downarrow E_w \mapsto \uparrow E_y) \quad (2.5)$$

Le nombre d'étapes dans chaque condition a diminué de n à 2.

- Si C_1 contient des contraintes fusionnées, $E_{CM} + E_{NCM1} + E_{NCM2} + \dots + E_{NCMi} \leq 1$

Il faut que la désactivation de l'étape commune $\downarrow E_{CM}$ permette l'activation des étapes non communes et vice versa. Elle sera donc exprimée par :

$$E_{CM} + E_{NCM1} + E_{NCM2} + \dots + E_{NCMi} \leq 1 \Leftrightarrow (\downarrow E_{CM} \mapsto \uparrow E_{NCM1} \vee \uparrow E_{NCM2} \vee \dots \vee \uparrow E_{NCMi}) \wedge (\downarrow E_{NCM1} \mapsto \uparrow E_{CM}) \wedge (\downarrow E_{NCM2} \mapsto \uparrow E_{CM}) \wedge \dots \wedge (\downarrow E_{NCMi} \mapsto \uparrow E_{CM}) \quad (2.6)$$

Réduire le nombre de conditions et le nombre d'étapes dans chaque condition revient à déterminer pour chaque étape commune E_{CM} , en utilisant le *GEAS*, l'étape prédécesseur $E_{Pré}$ et successeur E_{Suc} depuis les étapes E_{N-CM} qui constituent la contrainte (sachant que les bouclages seront pris en considération) (cf. figure 2.2).

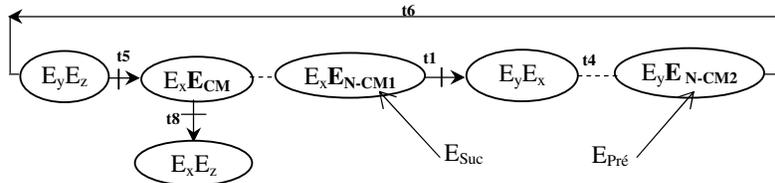


Figure 2.2 Détermination des étapes $E_{Pré}$ et E_{Suc} .
(Les étapes de la contrainte sont notées E_{CM} et E_{NCM})

¹ :Ce graphe représente le fonctionnement désiré du procédé.
Il est obtenu du GEA en supprimant tous les états frontières et interdits

L'ordre d'occurrence des étapes E_{N-CM} par rapport à E_{CM} permettra de simplifier la première condition de (2.6):

$$(\downarrow E_{CM} \mapsto \uparrow E_{NCM1} \vee \uparrow E_{NCM2} \vee \dots \vee \uparrow E_{NCMi}) \Leftrightarrow (\downarrow E_{CM} \mapsto \uparrow E_{Suc}) \text{ et de réduire les autres:}$$

$$(\downarrow E_{NCM1} \mapsto \uparrow E_{CM}) \wedge (\downarrow E_{NCM2} \mapsto \uparrow E_{CM}) \wedge \dots \wedge (\downarrow E_{NCMi} \mapsto \uparrow E_{CM}) \Leftrightarrow (\downarrow E_{Pré} \mapsto \uparrow E_{CM})$$

Pour aboutir à deux conditions chacune d'entre elles est constituée de deux étapes

$$E_{CM} + E_{Pré} + E_{Suc} + E_k + E_r \leq 1 \Leftrightarrow (\downarrow E_{Pré} \mapsto \uparrow E_{CM}) \wedge (\downarrow E_{CM} \mapsto \uparrow E_{Suc}) \quad (2.7)$$

b.Simplification du superviseur

La réduction des conditions qui assurent le respect des contraintes de l'ensemble C_1 (fusionnées ou non) aboutit à la diminution du nombre de conditions de supervision. Une deuxième simplification peut avoir lieu si certaines conditions sont vérifiées.

- Cas des contraintes fusionnées $(\downarrow E_{Pré} \mapsto \uparrow E_{CM}) \wedge (\downarrow E_{CM} \mapsto \uparrow E_{Suc})$

Les conditions de supervision permettent d'interdire le passage d'un état admissible qui inclut $E_{Pré}$ vers un état frontière qui inclut E_{CM} , ou d'un état admissible qui inclut E_{CM} vers un état frontière qui inclut E_{Suc} . L'interdiction de ces passages implique le blocage des états qui mettent le système dans des états frontières ou interdits.

➤ Si depuis un état admissible qui inclut $E_{Pré}$ on peut passer à un état frontière qui inclut E_{CM} (condition de précédence $C_{Pré}$), alors on conserve la condition de supervision $\downarrow E_{Pré} \mapsto \uparrow E_{CM}$. Ceci va donner naissance à une étape de contrôle dont l'entrée est $E_{Pré}$ et la sortie est E_{CM} .

➤ Si depuis un état admissible qui inclut E_{CM} on peut passer à un état frontière qui inclut E_{Suc} (condition de succession C_{Suc}), alors on va conserver la condition de supervision $\downarrow E_{CM} \mapsto \uparrow E_{Suc}$, ce qui va donner naissance à une étape de contrôle dont l'entrée est E_{CM} et la sortie est E_{Suc} (c.f figure 2.3).

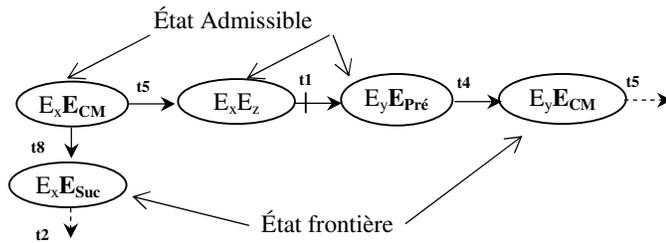


Figure 2.3 Conditions de supervision vérifiées sur le GEA.

Dans le cas contraire, la condition de supervision est inutile et sera donc supprimée.

- Cas des contraintes non fusionnées $(\downarrow E_y \mapsto \uparrow E_x) \wedge (\downarrow E_x \mapsto \uparrow E_z) \wedge \dots \wedge (\downarrow E_w \mapsto \uparrow E_y)$

Chaque contrainte non fusionnée est exprimée par un ensemble de conditions sous la forme de $(\downarrow E_i \mapsto \uparrow E_j)$. Chaque condition donnera naissance à une étape de contrôle si depuis un état admissible qui inclut E_i (étape désactivée) on peut passer à un état frontière qui inclut E_j (étape activée). L'entrée d'étape de contrôle sera E_i et la sortie E_j .

c. Recherche de l'étape initiale

Une condition de supervision vérifiée implique une étape de contrôle ajoutée au superviseur. Comme indiqué précédemment, la simplification des conditions de supervision permet d'une part de réduire le nombre d'étapes de contrôle, et d'autre part de limiter le nombre d'arcs entrant ou sortant d'une étape de contrôle à un seul arc.

Parmi les étapes de contrôle on doit avoir une étape qui sera initialement active, sinon le système sera bloqué. On recherche dans le *GEAS* du procédé non supervisé (sans état frontière ni état interdit) la première occurrence d'une étape de type E_{CM} ou $E_{Activée}$. L'étape de contrôle qui l'active (et donc qui la précède) sera considérée comme étape initiale.

d. Vérification du fonctionnement du système en boucle fermée

On doit vérifier que le modèle obtenu traduit bien le cahier de charges.

Pour cela on réalise d'abord le *GEA* du procédé supervisé à partir du *Grafcet* de commande avec les étapes de contrôle. Puis on vérifie que ce *GEA* ne comporte ni état frontière ni état interdit.

En effet le superviseur a pour rôle d'influer sur le fonctionnement du système en interdisant les états frontières et interdits. Par conséquent les états interdits et leurs arcs amont et aval seront automatiquement supprimés et les spécifications de fonctionnement seront ainsi satisfaites.

II.2. Application de la méthode de précedence sur le procédé SPT

On va appliquer la technique de supervision développée sur l'exemple donné figure 2.4. Le procédé est constitué de deux sous systèmes qui assurent le pesage de produits puis leur transport.

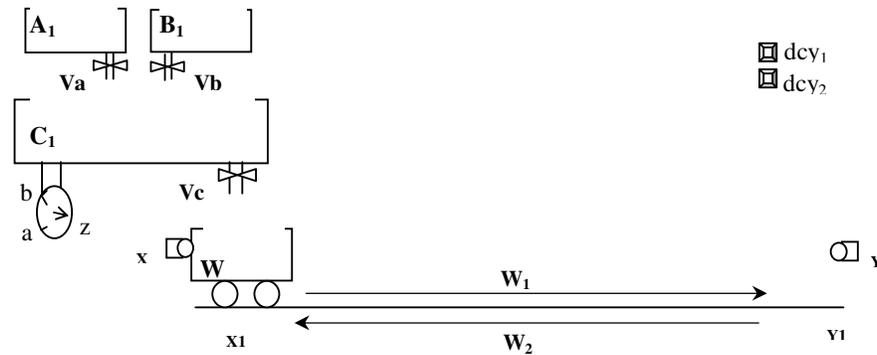


Figure 2.4 Système de Pesage et Transport (SPT).

A l'état initial, la cuve C_1 est vide (zéro z de la bascule), et le wagon est en position gauche (détecté par x).

Le cycle de travail consiste à peser successivement les deux produits A et B jusqu'aux niveaux a et b.

Si le wagon est dans la position $X1$ on peut vider la cuve C_1 .

Dès que le niveau z de la balance est atteint, le wagon W sera déplacé vers la position $Y1$ par l'action de la commande W_1 . Dans cette position le wagon sera vidé par l'action V_{in} (on ne s'intéresse pas au détail de la vidange).

Après la fin de la vidange (détectée par v_i) la commande W_2 permet le retour du wagon à sa position initiale.

Les événements b , $\uparrow x$, v_i sont incontrôlables.

II.2.a. Graphes d'états du SPT

II.2.a.1. Grafquets de commande et de synchronisation du SPT

Les commandes du système de pesage et de transport sont modélisées séparément, puis on effectue la synchronisation.

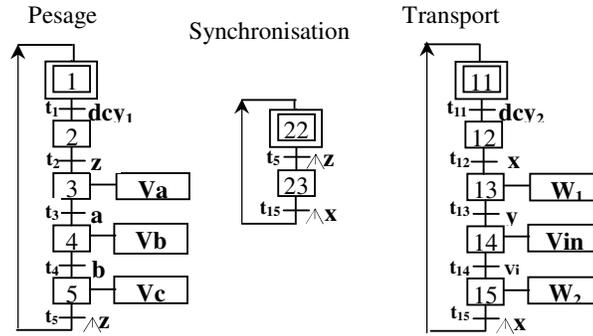


Figure 2.5 Grafcets de commande du SPT.

Les transitions à événements contrôlables sont

$$t_1=dcy_1, t_2=\uparrow z, t_3=a, t_5=z, t_{11}=dcy_2, t_{13}=y.$$

Les transitions à événements incontrôlables sont

$$t_4=b, t_{12}=\uparrow x, t_{14}=vi, t_{15}=x.$$

La synchronisation se fait grâce aux événements $\uparrow z$ et $\uparrow x$ (z et x sont des conditions initiales). L'ouverture de la vanne (commande Vc) de la cuve C₁ ne peut avoir lieu que si le wagon W est en position gauche, détectée par $\uparrow x$. La commande de déplacement à droite du wagon (W₁) ne peut se faire que si la cuve C₁ est vide, détectée par le zéro de la bascule $\uparrow z$.

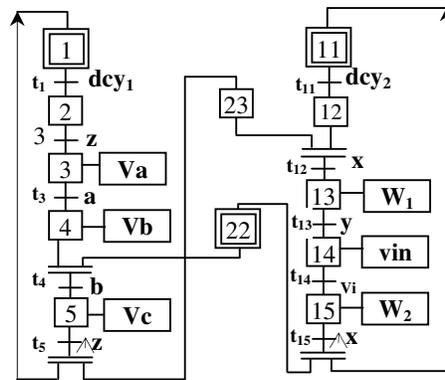


Figure 2.6 Grafcet du procédé (SPT) non supervisé.

II.2.a.2. Graphe d'états accessibles du SPT

Le graphe d'états accessibles décrit les phases successives par lesquelles passera le système.

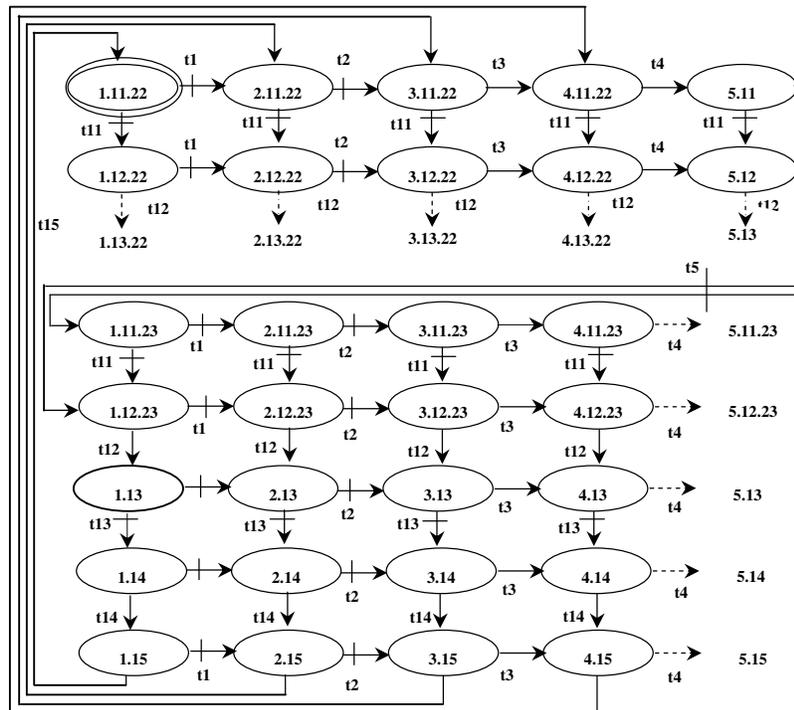


Figure 2.7 Graphe d'états accessibles du procédé (SPT) non supervisé.

II.2.a.3. Extraction des anomalies

Commençons par définir les ensembles des états frontières E_B , interdits E_I et admissibles E_A . Ils sont déduits directement du graphe d'états accessibles:

$$E_A = \{E_1E_{11}E_{22}, E_2E_{11}E_{22}, E_3E_{11}E_{22}, E_4E_{11}E_{22}, E_5E_{11}, E_1E_{11}E_{23}, E_2E_{11}E_{23}, E_1E_{12}E_{23}, E_2E_{12}E_{23}, E_1E_{13}, E_2E_{13}, E_1E_{14}, E_2E_{14}, E_1E_{15}, E_2E_{15}\}. \tag{2.8}$$

$$E_I = \{E_1E_{12}E_{22}, E_2E_{12}E_{22}, E_3E_{12}E_{22}, E_4E_{12}E_{22}, E_5E_{12}, E_4E_{11}E_{23}, E_4E_{12}E_{23}, E_4E_{13}, E_4E_{14}, E_4E_{15}\}. \tag{2.9}$$

$$E_B = \{E_1E_{12}E_{22}, E_2E_{12}E_{22}, E_3E_{12}E_{22}, E_4E_{12}E_{22}, E_5E_{12}, E_3E_{11}E_{23}, E_3E_{12}E_{23}, E_3E_{13}, E_3E_{14}, E_3E_{15}\}. \tag{2.10}$$

II.2.b. Synthèse du superviseur du procédé (SPT)

II.2.b.1. Réduction du nombre de contraintes linéaire

- On exprime les états frontières sous forme de contraintes linéaires

$$\begin{aligned}
 &E_1 + E_{12} + E_{22} \leq 2 \text{ (au maximum deux étapes actives sur trois)} \\
 &E_2 + E_{12} + E_{22} \leq 2 \\
 &E_3 + E_{12} + E_{22} \leq 2 \\
 &E_4 + E_{12} + E_{22} \leq 2 \\
 &E_5 + E_{12} \leq 1 \text{ (au maximum une étape active sur deux)} \\
 &E_3 + E_{11} + E_{23} \leq 2 \\
 &E_3 + E_{12} + E_{23} \leq 2 \\
 &E_3 + E_{13} \leq 1 \\
 &E_3 + E_{14} \leq 1 \\
 &E_3 + E_{15} \leq 1
 \end{aligned}
 \tag{2.11}$$

- *Construction de l'ensemble de sur états frontières et admissibles* \mathcal{B}_1 et \mathcal{A}_1

On a 10 états frontières notés b_1 à b_{10} . $b_1 = E_1E_{12}E_{22}$, $b_2 = E_2E_{12}E_{22}$, ..., $b_{10} = E_3E_{15}$.

L'ensemble des *sur états* de b_1 est: $b_1^{sur} = \{E_1, E_{12}, E_{22}, E_1E_{12}, E_1E_{22}, E_{12}E_{22}, E_1E_{12}E_{22}\}$.

L'ensemble des *sur états* de b_2 est: $b_2^{sur} = \{E_2, E_{12}, E_{22}, E_2E_{12}, E_2E_{22}, E_{12}E_{22}, E_2E_{12}E_{22}\}$.

.....

L'ensemble des *sur états* de b_{10} est : $b_{10}^{sur} = \{E_3, E_{15}, E_3E_{15}\}$.

L'ensemble des *sur états* frontière est donné par $B_1 = \bigcup_{i=1}^{10} b_i^{sur}$, $\mathcal{B}_1 = \{b_1^{sur}, b_2^{sur}, \dots, b_{10}^{sur}\}$.

$\mathcal{B}_1 = \{E_1, E_2, E_3, E_4, E_5, E_{11}, E_{12}, E_{13}, E_{14}, E_{15}, E_{22}, E_{23}, E_1E_{12}, E_1E_{22}, E_2E_{12}, E_2E_{22}, E_3E_{12}, E_3E_{22}, E_4E_{12}, E_4E_{22}, E_{12}E_{22}, E_5E_{12}, E_3E_{13}, E_3E_{14}, E_3E_{15}, E_3E_{11}, E_{11}E_{23}, E_{12}E_{23}, E_3E_{23}, E_1E_{12}E_{22}, E_2E_{12}E_{22}, E_3E_{12}E_{22}, E_4E_{12}E_{22}, E_3E_{11}E_{23}, E_3E_{12}E_{23}\}$.

(2.12)

On a 15 états admissibles notés a_1 à a_{15} . $a_1 = E_1E_{11}E_{22}$, $a_2 = E_2E_{11}E_{22}$, ..., $a_{15} = E_2E_{15}$.

L'ensemble des *sur états* de a_1 est: $a_1^{sur} = \{E_1, E_{11}, E_{22}, E_1E_{11}, E_1E_{22}, E_{11}E_{22}, E_1E_{11}E_{22}\}$.

L'ensemble des *sur états* de a_2 est: $a_2^{sur} = \{E_2, E_{11}, E_{22}, E_2E_{11}, E_2E_{22}, E_{11}E_{22}, E_2E_{11}E_{22}\}$.

...

L'ensemble des *sur états* de a_{15} est: $a_{15}^{sur} = \{E_2, E_{15}, E_2E_{15}\}$.

L'ensemble des *sur états* admissibles est donné par $A_1 = \bigcup_{i=1}^{15} a_i^{sur}$, $\mathcal{A}_1 = \{a_1^{sur}, a_2^{sur}, \dots, a_{15}^{sur}\}$.

$\mathcal{A}_1 = \{E_1, E_2, E_3, E_4, E_5, E_{11}, E_{12}, E_{13}, E_{14}, E_{15}, E_{22}, E_{23}, E_1E_{11}, E_1E_{22}, E_2E_{11}, E_2E_{22}, E_3E_{11}, E_3E_{22}, E_4E_{11}, E_4E_{22}, E_{11}E_{22}, E_5E_{11}, E_1E_{11}, E_1E_{23}, E_2E_{11}, E_2E_{23}, E_{11}E_{23}, E_1E_{12}, E_2E_{12}, E_{12}E_{23}, E_1E_{13}, E_2E_{13}, E_1E_{14}, E_2E_{14}, E_1E_{15}, E_2E_{15}, E_1E_{11}E_{22}, E_2E_{11}E_{22}, E_3E_{11}E_{22}, E_4E_{11}E_{22}, E_1E_{11}E_{23}, E_2E_{11}E_{23}, E_1E_{12}E_{23}, E_2E_{12}E_{23}\}$.

(2.13)

- *Suppression des sur états admissibles de l'ensemble de sur états frontières*

Il faut supprimer les *sur états* admissibles inclus dans \mathcal{B}_2 pour éviter leurs interdiction, $\mathcal{B}_2' = \mathcal{B}_2 / \mathcal{A}_1$.

$\mathcal{B}_2' = \{E_3E_{12}, E_3E_{13}, E_3E_{14}, E_3E_{15}, E_3E_{23}, E_4E_{12}, E_5E_{12}, E_{12}E_{22}, E_1E_{12}E_{22}, E_2E_{12}E_{22}, E_3E_{12}E_{22}, E_4E_{12}E_{22}, E_3E_{11}E_{23}, E_3E_{12}E_{23}\}$.

(2.14)

- *Suppression des sur états non minimaux*

Les *sur états* redondants de la liste \mathcal{B}_2' seront supprimés afin d'obtenir l'ensemble final des *sur états* frontières :

$$\mathcal{B}_2 = \{ E_3E_{12}, E_3E_{13}, E_3E_{14}, E_3E_{15}, E_3E_{23}, E_4E_{12}, E_5E_{12}, E_{12}E_{22} \}. \quad (2.15)$$

- *Choix des sur états qui couvrent un maximum de contraintes linéaires*

Tableau 2.1 Table de réduction sur états – contraintes.

$\mathcal{B}_2 \setminus \mathcal{B}_1$	$E_2E_{12}E_{22}$	$E_4E_{12}E_{22}$	$E_3E_{11}E_{23}$	E_3E_{13}	E_3E_{15}
	$E_1E_{12}E_{22}$	$E_3E_{12}E_{22}$	E_5E_{12}	$E_3E_{12}E_{23}$	E_3E_{14}
E_3E_{12}		/		/	
E_3E_{13}					/
E_3E_{14}					/
E_3E_{15}					/
E_3E_{23}			/	/	
E_4E_{12}		/			
E_5E_{12}			/		
$E_{12}E_{22}$	/	/			

Le tableau 2.1 permet de définir l'ensemble réduit de *sur états* :

$$C_2 = \{ E_3E_{13}, E_3E_{14}, E_3E_{15}, E_3E_{23}, E_5E_{12}, E_{12}E_{22} \}. \quad (2.16)$$

- *Fusionnement des contraintes par recherche d'éléments communs*

On a deux ensembles de contraintes linéaires.

Le premier est donné par: $E_3 + E_{13} \leq 1, E_3 + E_{14} \leq 1, E_3 + E_{15} \leq 1, E_3 + E_{23} \leq 1$

L'étape 3 est un élément commun E_{CM} pour cet ensemble de contraintes. Comme

$\{E_{13}E_{14}, E_{13}E_{15}, E_{13}E_{23}, E_{14}E_{15}, E_{14}E_{23}, E_{15}E_{23}\} \notin A1$, on peut alors remplacer les contraintes par :

$$E_3 + E_{13} + E_{14} + E_{15} + E_{23} \leq 1. \quad (2.17)$$

Le deuxième ensemble est donné par $E_{12} + E_5 \leq 1, E_{12} + E_{22} \leq 1$.

L'étape 12 est un élément commun E_{CM} pour les deux contraintes.

Comme $\{E_5E_{22}\} \notin A1$ alors on remplace les deux contraintes par:

$$E_{12} + E_5 + E_{22} \leq 1. \quad (2.18)$$

L'ensemble de *sur états* frontières après fusionnement est :

$$C_1 = \{ (E_3, E_{13}E_{14}E_{15}E_{23}), (E_{12}, E_5E_{22}) \}. \quad (2.19)$$

Remarque : Si les éléments de C_2 sont indépendants les uns des autres (pas d'étape commune), alors $C_1 = C_2$.

II.2.b.2. Construction du superviseur du SPT "principe de précedence"

En premier lieu il faut construire le "graphe d'états accessibles simplifié" *GEAS* par la suppression des états interdits et frontières du graphe d'états accessibles *GEA*.

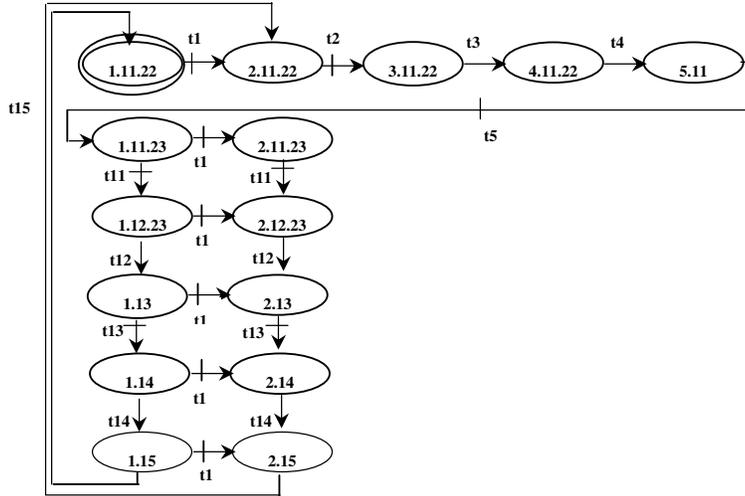


Figure 2.8 Graphe d'états accessibles simplifié du procédé (SPT) non supervisé.

Ensuite pour chaque contrainte, on localise l'étape E_{CM} ainsi que $E_{Pré}$ et E_{Suc} avec la prise en considération des bouclages.

Pour la première contrainte $E_3 + E_{13} + E_{14} + E_{15} + E_{23} \leq 1$, l'étape commune est E_3 .

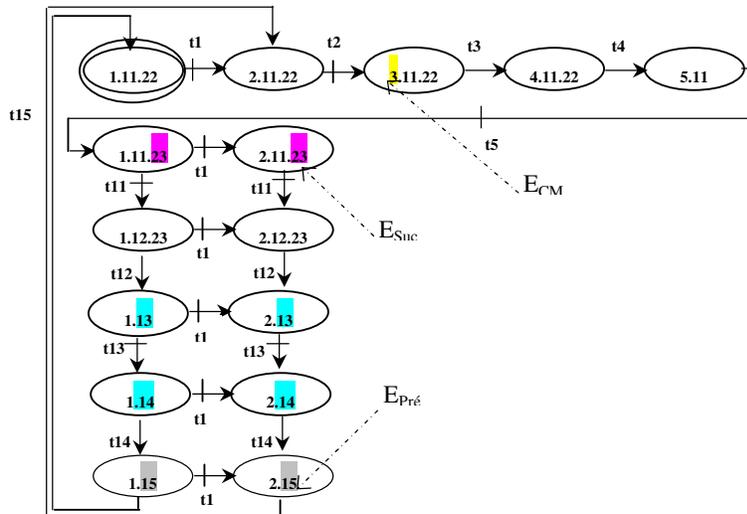


Figure 2.9 $E_{Pré}$ et E_{Suc} de l'étape E_3 sur le GEAS.

De la figure 2.9, on voit que l'étape qui précède E_3 est E_{15} et celle qui lui succède est E_{23} . Par conséquent la désactivation de E_{15} permet l'activation de E_3 , et la désactivation de E_3 permet l'activation de E_{23} . On peut exprimer cela par :

$$(\downarrow E_{15} \mapsto \uparrow E_3) \wedge (\downarrow E_3 \mapsto \uparrow E_{23}). \tag{2.20}$$

Pour la deuxième contrainte $E_{12} + E_5 + E_{22} \leq 1$, on recherche $E_{Pré}$ et E_{Suc} de l'étape commune E_{12} .

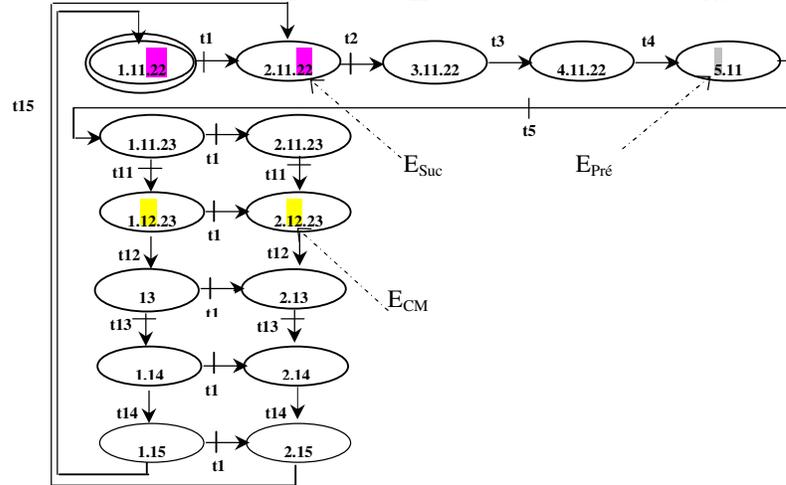


Figure 2.10 $E_{Pré}$ et E_{Suc} de l'étape E_{12} sur le GEAS.

L'étape qui précède E_{12} est E_5 et celle qui lui succède est E_{22} . Par conséquent on peut écrire : $(\downarrow E_5 \mapsto \uparrow E_{12}) \wedge (\downarrow E_{12} \mapsto \uparrow E_{22})$. (2.21)

Enfin on vérifie sur le graphe d'états accessibles les conditions de supervision. Pour le groupe d'étape $E_{15}E_3E_{23}$ on vérifie si depuis un état admissible qui inclut l'étape 15 on peut passer à un état frontière qui inclut l'étape 3. Et si depuis un état admissible qui inclut l'étape 3 on peut passer à un état frontière qui inclut l'étape 23. On doit faire la même vérification pour le groupe d'étapes $E_5E_{12}E_{22}$.

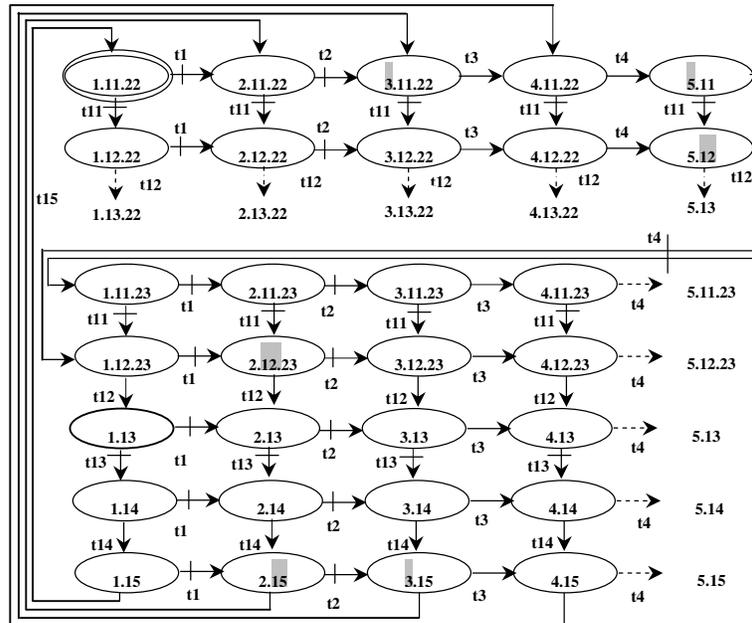


Figure 2.11 Conditions de supervision sur le GEA du SPT.

Sur la figure 2.11, on observe que l'on a deux conditions vérifiées sur quatre : entre les étapes 15-3 et 5-12. Par conséquent on a deux étapes de contrôle EC_{153} , EC_{512} . Donc l'entrée de EC_{153} sera E_{15} et sa sortie sera E_3 . De même l'entrée de EC_{512} sera E_5 et la sortie sera E_{12} .

Les conditions non vérifiées n'ont aucune influence sur le superviseur.

II.2.c. Fonctionnement du procédé SPT supervisé

II.2.c.1. Construction du superviseur

Pour trouver l'étape initiale du superviseur, on cherche d'abord parmi les étapes communes E_3 et E_{12} celle qui est activée en premier dans le *GEAS*.

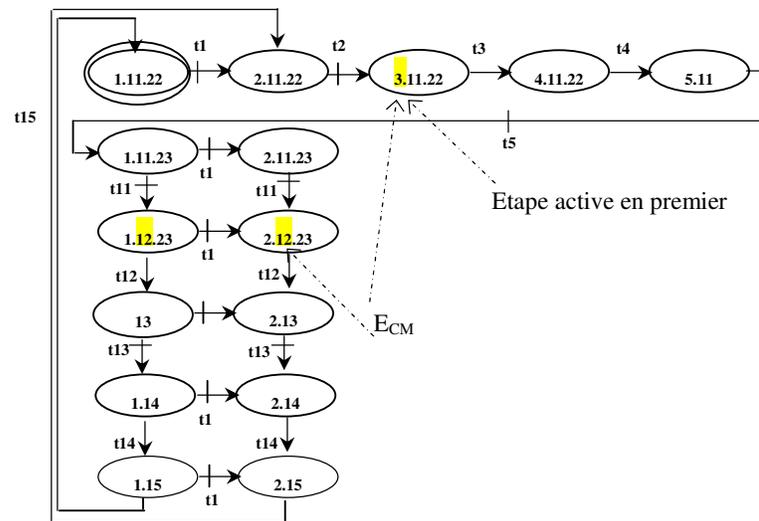


Figure 2.12 Graphe d'Etats Accessibles Simplifié du SPT.

D'après le graphe de la figure 2.12, on observe que l'étape commune E_3 est active avant l'étape commune E_{12} . Par conséquent l'étape de contrôle E_{153} qui permet d'activer l'étape E_3 sera considérée comme étape initiale.

Le superviseur qui permet de satisfaire les contraintes linéaires données par la relation (2.11) est finalement exprimé par deux étapes de contrôle: E_{153} et E_{512} . Elles seront rajoutées sur le *Grafcet* initial, pour obtenir le *Grafcet* de commande du procédé supervisé (cf figure 2.13).

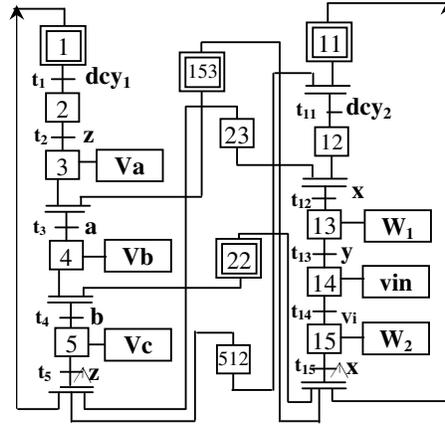


Figure 2.13 Grafcet de commande du procédé (SPT) supervisé.

II.2.c.2. Vérification du fonctionnement du SPT en boucle fermée

A partir du *Grafcet* de la figure 2.13, on construit le graphe d'états accessibles du procédé supervisé.

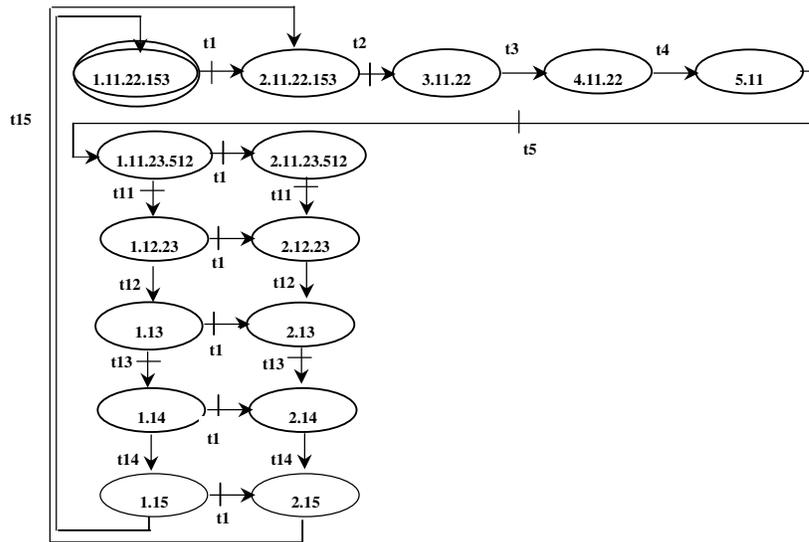


Figure 2.14 Graphe d'états accessibles sous superviseur du SPT.

On remarque que le graphe d'états accessibles avec superviseur ne comporte ni état frontière ni état interdit. Donc les spécifications de fonctionnement sont satisfaites.

En effet, dans le *Grafcet* supervisé l'étape de contrôle 153 garantit que le passage de l'étape E_4 à E_5 ne peut être effectif que si E_{22} est active, car l'activité de E_4 est subordonnée au franchissement de la transition contrôlable t_3 , contrôlé par l'étape 153 (activée en même temps que E_{22}). Par conséquent la vidange de la cuve C_1 ne peut se faire que si le wagon est situé à la position X_1 .

Il en est de même pour le déplacement du wagon avant la fin du vidange. Il est interdit grâce à l'étape de contrôle 512. L'étape de contrôle qui agit en amont de l'étape de synchronisation nous garantit que le franchissement de la transition incontrôlable se produira toujours quand l'étape de synchronisation est active. Par conséquent le travail de cette dernière devient inutile sur le plan structurel. On peut donc supprimer cette étape de synchronisation sans aucune incidence sur le fonctionnement du *Grafcet* de commande supervisé.

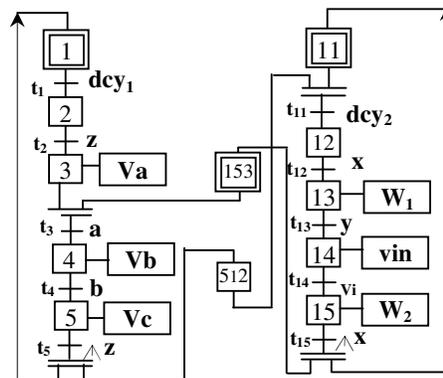


Figure 2.15 Grafcet de commande simplifié du procédé (SPT) supervisé.

II.3. Application de la méthode structurale sur le procédé SPT

Compte tenu du fait que la méthode de synthèse structurale développée par *Kattan* [12] est très proche de la notre dans sa partie analyse, appliquons cette technique sur l'exemple déjà utilisé pour pouvoir comparer les résultats avec les notre.

Avant de procéder à l'application on va résumer les étapes principales de l'approche utilisée par *Kattan*.

1. La commande et la synchronisation seront modélisées par le *Grafcet*, les transitions du *Grafcet* de commande et de synchronisation qui possèdent le même événement seront couplées

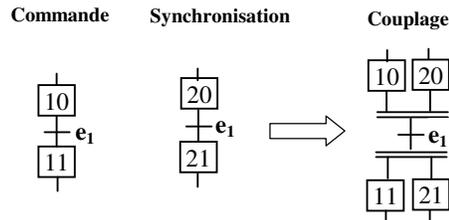


Figure 2.16 Couplage de la synchronisation avec la commande.

2. Réalisation du *quasi-Grafcet* qui représente le fonctionnement obtenu depuis le *Grafcet* de commande du procédé, il est similaire au graphe d'états accessibles.
3. Définition des états frontières qui seront transformés en contraintes linéaires.
4. Réduction des contraintes par la méthode de l'*invariant*.
5. Détermination des étapes de contrôle depuis la matrice du procédé ainsi que les états initiaux des étapes de contrôle.
6. Établissement du *Grafcet* supervisé.

Application de la technique de Kattan

- Si on prend le procédé de la figure 2.4 et qu'on lui applique la première étape pour fusionner les *Grafcets* de commande et de synchronisation de la figure 2.5, on obtient :

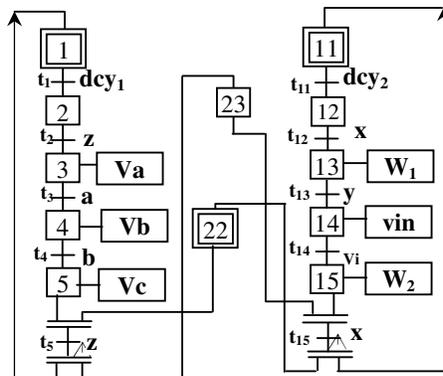


Figure 2.17 Grafcet de commande du procédé avec synchronisation.

Cette méthode de synchronisation, valable pour les réseaux de *Pétri* et pour certains *Grafquets*, n'est pas valable dans le cas général et en particulier pour notre exemple, car le *Grafcet* du procédé étendu obtenu ne satisfait pas les exigences de fonctionnement imposées dans l'énoncé.

Pour cela, dans toute la suite on va utiliser le *Grafcet* de commande global de la figure 2.6, obtenu avec notre technique de synchronisation.

- Le *quasi-Grafcet* obtenu est semblable au graphe d'états accessibles de la figure 2.7, où les états admissibles et frontières sont donnés par les relations (2.8) et (2.10).

- Les contraintes linéaires auxquelles on aboutit depuis les états frontières sont exprimées par la relation (2.11) :

$$\begin{array}{cccccc} E_1+E_{12}+E_{22} \leq 2 & E_2+E_{12}+E_{22} \leq 2 & E_3+E_{12}+E_{22} \leq 2 & E_4+E_{12}+E_{22} \leq 2 & E_5+E_{12} \leq 1 \\ E_3+E_{11}+E_{23} \leq 2 & E_3+E_{12}+E_{23} \leq 2 & E_3+E_{13} \leq 1 & E_3+E_{14} \leq 1 & E_3+E_{15} \leq 1 \end{array}$$

II.3.a. Réduction des contraintes par la méthode de l'invariant

L'application de la technique de l'*invariant* [8] permet de réduire les contraintes linéaires.

On groupe les contraintes dont le nombre d'étapes est supérieur à 2 et possédant des étapes communes. Le premier groupe est :

$$\left. \begin{array}{l} E_1+E_{12}+E_{22} \leq 2 \\ E_2+E_{12}+E_{22} \leq 2 \\ E_3+E_{12}+E_{22} \leq 2 \\ E_4+E_{12}+E_{22} \leq 2 \end{array} \right\} \quad (2.22)$$

Les étapes E_1, E_2, E_3, E_4 ne sont jamais actives en même temps, elles forment un *invariant* [15]. Par conséquent on a $E_1 + E_2 + E_3 + E_4 = 1$. Cela permet de remplacer les quatre contraintes linéaires du dessus par une seule : $E_{12} + E_{22} \leq 1$.

De même le deuxième groupe de contraintes donné par :

$$\left. \begin{array}{l} E_3+E_{11}+E_{23} \leq 2 \\ E_3+E_{12}+E_{23} \leq 2 \end{array} \right\} \quad (2.23)$$

Est remplacé par $E_3 + E_{23} \leq 1$, car les étapes E_{11} et E_{12} représentent un *invariant* ($E_{11} + E_{12} = 1$).

La simplification des contraintes permet de réduire le nombre de contraintes de 10 à 6:

$$\left. \begin{array}{l} E_{12}+E_{22} \leq 1 \\ E_3+E_{23} \leq 1 \\ E_5+E_{12} \leq 1 \\ E_3+E_{13} \leq 1 \\ E_3+E_{14} \leq 1 \\ E_3+E_{15} \leq 1 \end{array} \right\} \quad (2.24)$$

II.3.b. Détermination des étapes de contrôle

Les étapes de contrôle qu'il faut ajouter au procédé pour satisfaire les contraintes linéaires sont exprimées sous forme matricielle en fonction de la matrice du procédé $Wp(n.m)$ et la matrice de contraintes $L(l.n)$,

$$Wc(l.m) = -L(l.n) \cdot Wp(n.m) \quad (2.25)$$

où n :est le nombre d'étapes du procédé.

m :est le nombre de transitions.

l :est le nombre de contraintes.

La matrice $Wp(n.m)$ exprime la relation entre les étapes et les transitions du procédé non supervisé, les transitions sont mises en colonnes et les étapes en lignes. L'intersection des deux prend -1 si l'étape est en amont de la transition, et 1 dans le cas inverse. S'il n'y a pas de lien entre les deux on aura un 0.

La matrice $L(l.n)$ représente les étapes qui sont éléments des contraintes, les étapes sont mises en colonnes et les contraintes en lignes.

Chaque ligne de la matrice $Wc(l.m)$ permet d'identifier une étape de contrôle qui sera ajoutée au procédé pour établir la supervision.

Le procédé contient 12 étapes et 10 transitions.

$$W_{P(12,10)} = \begin{bmatrix} E_1 & -1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ E_2 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ E_3 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ E_4 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ E_5 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 \\ E_{11} & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 1 \\ E_{12} & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 \\ E_{13} & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 \\ E_{14} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 \\ E_{15} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 \\ E_{22} & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 1 \\ E_{23} & 0 & 0 & 0 & 0 & 1 & 0 & -1 & 0 & 0 & 0 \\ t_1 & t_2 & t_3 & t_4 & t_5 & t_{11} & t_{12} & t_{13} & t_{14} & t_{15} \end{bmatrix}$$

On a 6 contraintes :

$$L_{(6.12)} = - \begin{bmatrix} & E_1 & E_2 & E_3 & E_4 & E_5 & E_{11} & E_{12} & E_{13} & E_{14} & E_{15} & E_{22} & E_{23} \\ C_1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ C_2 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ C_3 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ C_4 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ C_5 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ C_6 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix}$$

Depuis L et Wp on construit la matrice Wc :

$$Wc_{(6.10)} = - L_{(6.12)} \cdot Wp_{(12.10)},$$

$$Wc_{(6.10)} = - \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} -1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & -1 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$Wc_{(6.10)} = \begin{bmatrix} & t_1 & t_2 & t_3 & t_4 & t_5 & t_{11} & t_{12} & t_{13} & t_{14} & t_{15} \\ Ec_{31} & 0 & 0 & 0 & 1 & 0 & -1 & 1 & 0 & 0 & -1 \\ Ec_{32} & 0 & -1 & 1 & 0 & -1 & 0 & 1 & 0 & 0 & 0 \\ Ec_{33} & 0 & 0 & 0 & -1 & 1 & -1 & 1 & 0 & 0 & 0 \\ Ec_{34} & 0 & -1 & 1 & 0 & 0 & 0 & -1 & 1 & 0 & 0 \\ Ec_{35} & 0 & -1 & 1 & 0 & 0 & 0 & 0 & -1 & 1 & 0 \\ Ec_{36} & 0 & -1 & 1 & 0 & 0 & 0 & 0 & 0 & -1 & 1 \end{bmatrix}$$

Selon la matrice $Wc_{(6.10)}$ on a 6 étapes de contrôle (E_{31}, \dots, E_{36}). Chacune a deux transitions en entrée et deux en sortie.

Les transitions qui ont -1 permettent de désactiver l'étape de contrôle correspondante, et celles qui ont 1 autorisent l'activation (t_{11} et t_{15} activent E_{c31} qui sera désactivée par t_4 et t_{12}).

II.3.b.1. Détermination des étapes initiales

Pour obtenir les étapes de contrôle qui seront initialement actives, il faut appliquer la formule suivante : $X_{cini}(l.I) = b(l.I) - L(l.n).X_{pini}(n.I)$ (2.26)

Où $X_{cini}(l.I)$: matrice d'état initiale des étapes de contrôle.

$b(l.I)$: matrice des bornes supérieures des contraintes linéaires.

$X_{pini}(n.I)$: matrice d'état initiale des étapes du procédé.

Les étapes de contrôle initialement actives sont données par :

$$X_{cini}(6.I) = b(6.I) - L(6.12).X_{pini}(12.I),$$

$$X_{cini} = \begin{bmatrix} b \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} - \begin{bmatrix} C_1 \\ C_2 \\ C_3 \\ C_4 \\ C_5 \\ C_6 \end{bmatrix} \begin{bmatrix} E_1 & E_2 & E_3 & E_4 & E_5 & E_{11} & E_{12} & E_{13} & E_{14} & E_{15} & E_{22} & E_{23} \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} E_1 & 1 \\ E_2 & 0 \\ E_3 & 0 \\ E_4 & 0 \\ E_5 & 0 \\ E_{11} & 1 \\ E_{12} & 0 \\ E_{13} & 0 \\ E_{14} & 0 \\ E_{15} & 0 \\ E_{22} & 1 \\ E_{23} & 0 \end{bmatrix}$$

$$X_{cini} = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} - \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

$$X_{cini} = \begin{bmatrix} Ec_{31} & 0 \\ Ec_{32} & 1 \\ Ec_{33} & 1 \\ Ec_{34} & 1 \\ Ec_{35} & 1 \\ Ec_{36} & 1 \end{bmatrix}$$

Donc les étapes de contrôle 32, 33, 34, 35, 36 seront initialement actives.

II.3.c. Etablissement du Grafcet supervisé

On implémente les étapes de contrôle (le superviseur) sur le *Grafcet* de commande.

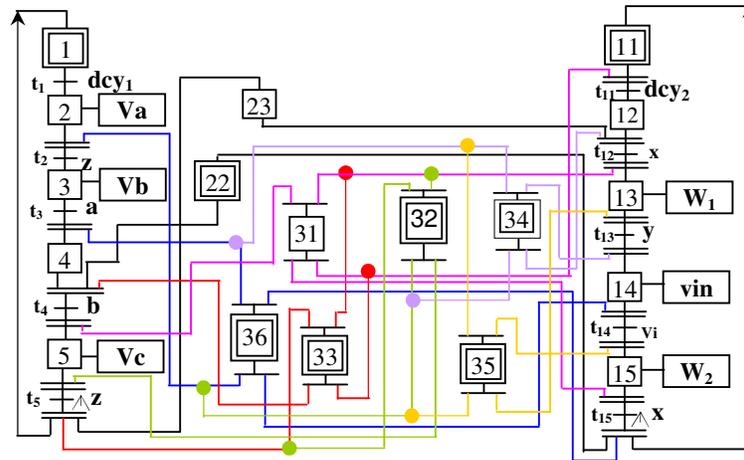


Figure 2.18 Grafcet de commande du SPT avec le superviseur de Kattan.

II.4. Comparaison des résultats obtenus par les deux méthodes

La première différence entre les deux méthodes est la manière de synchroniser les différents *Grafcets*. *Kattan* a utilisé la même méthode de synchronisation que celle appliquée pour les *RDP*, qui se base uniquement sur les événements de fin d'action. Cette technique n'est pas valable pour la modélisation par *Grafcet*, car la synchronisation doit se faire avec les événements qui indiquent l'achèvement des actions et ceux qui donnent l'ordre de démarrage des autres actions.

Selon l'exemple l'étape 22 doit synchroniser le fonctionnement du procédé entre la présence du wagon à la position X1 et l'ouverture de la vanne V_C . Mais par la représentation de *Kattan* la vanne est ouverte avant car la synchronisation sera liée à $\uparrow z$ qui indique la fin de la vidange de la cuve C_1 .

De même l'événement $\uparrow z$ doit permettre le déplacement du wagon vers Y1, et non pas s'assurer que le wagon est à la position X1 à la fin de la vidange.

D'autre part on remarque que le nombre d'étapes de contrôle obtenu par l'application de la technique de *Kattan* est important (6 étapes de contrôle) cf figure 2.18, chacune possédant 4 arcs. L'activation et la désactivation sont effectuées par le biais de deux arcs. Par contre dans notre méthode on a obtenu un superviseur constitué seulement de deux étapes de contrôle (cf figure 2.15), de plus le nombre d'arcs dédié à chaque étape de contrôle est limité à deux, un pour l'activation et l'autre pour la désactivation.

II.5.Conclusion

Dans notre technique les étapes de synchronisation qui agissent sur un événement incontrôlable peuvent être supprimées en toute sécurité, car leur rôle est accompli par le superviseur, alors que dans la technique de *Kattan* les étapes de synchronisation ont été maintenues dans le *Grafcet* de commande supervisé, ce qui diminue la lisibilité.

La technique développée a permis de synthétiser un superviseur optimal. En effet grâce au principe de précedence on a eu deux réductions :

- La réduction du nombre des conditions a impliqué une réduction du nombre d'étapes de contrôle.
- La réduction du nombre d'étapes qui constituent chaque condition a permis la réduction du nombre d'arcs liés à chaque étape de contrôle.

D'autre part les étapes de contrôle qui ne vérifient pas les conditions de précedence, seront supprimées car inutiles et n'apportant pas un plus dans le fonctionnement.

La taille réduite du superviseur obtenu permettra de faciliter l'implémentation sur l'automate programmable industriel.

CHAPITRE III

*Exemple d'implémentation
sur API*

Introduction

Ce chapitre est consacré à l'implémentation sur automate programmable d'un *Grafset* de supervision. Le procédé correspond au poste d'alimentation, matriçage et évacuation (AME) d'une installation électropneumatique.

Après avoir appliqué la technique développée au chapitre précédent pour synthétiser le superviseur de commande, on effectue l'implémentation en langage en PL7-2 sur l'automate programmable industriel TSX17-20.

III.1. Description et principe de fonctionnement du procédé AME

Le procédé AME est formé de deux sous systèmes. Le premier dédié à l'alimentation du poste en pièces brutes par le robot, et le second au matriçage et évacuation des pièces formatées.

Les composants constituant les deux entités sont :

- Un magasin d'alimentation en pièces brutes où elles descendent par gravité,
- Un robot électropneumatique équipé de vérins et de capteurs de fin de course de diverses technologies.
- Deux vérins pour le matriçage et l'évacuation des pièces.
- Un bac de réception des pièces formatées.
- Une platine d'interface sur laquelle sont montés les composants électropneumatiques (modules d'interfaces électropneumatiques, distributeurs, sectionneur, etc..)
- Une boîte à bouton pour le dialogue homme-machine.
- Un API de type TSX17-20 avec son environnement de programmation PL7-2.
- La console de programmation T407 avec la cartouche de l'interpréteur du langage.
- Une boîte à boutons pour le dialogue homme-machine.

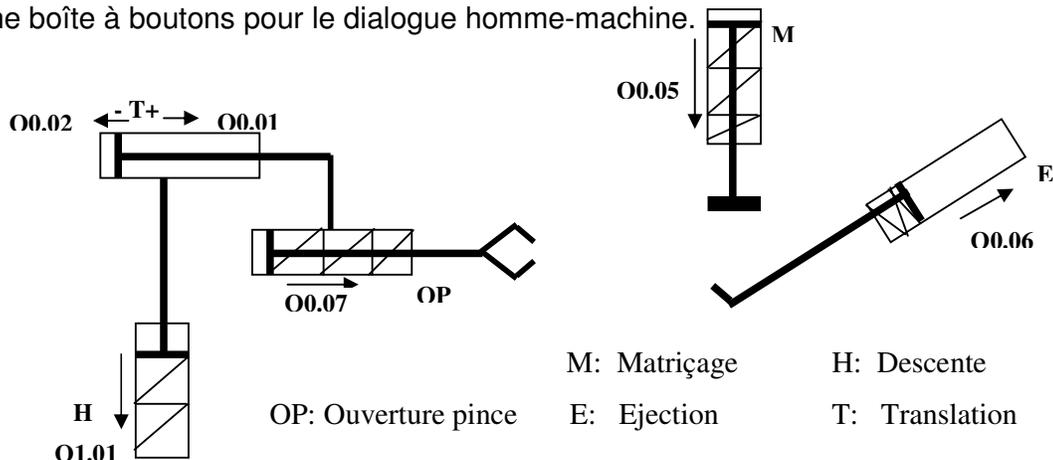


Figure 3.1 Identification des actionneurs.

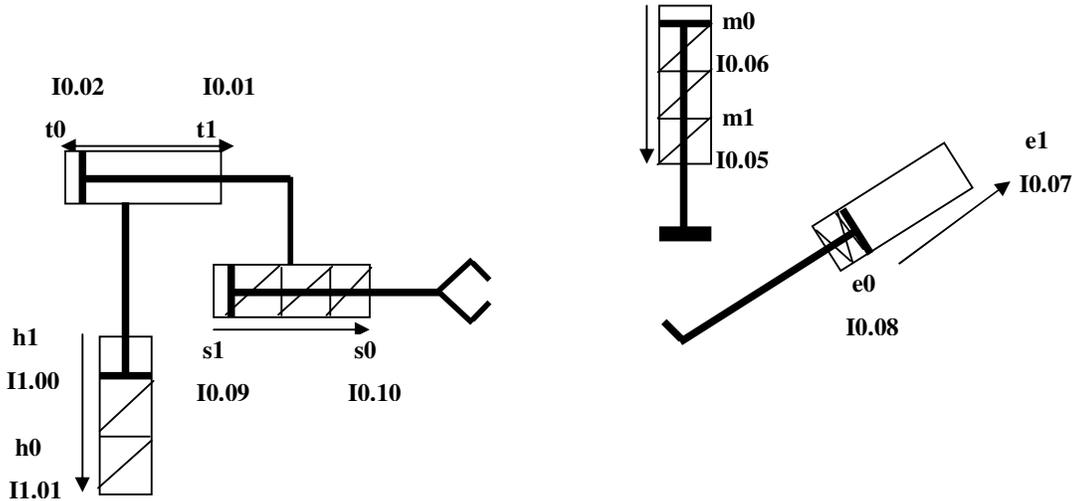


Figure 3.2 Identification des détecteurs de fin de course.

Un appui sur le bouton poussoir dcy donne le départ cycle aux deux sous systèmes. La séquence des opérations (cf figure 3.3) donne le cycle de fonctionnement du procédé global.

L'opération d'alimentation comprend 7 phases :

- ouverture de la pince par le vérin OP (commande du vérin OP)
- descente du bras du robot (commande du vérin H)
- fermeture de la pince (retour par ressort de rappel du vérin OP)
- montée du bras (retour par ressort de rappel du vérin H)
- transfert de la pièce par translation du bras (commande T+ du vérin T)
- ouverture pince (commande du vérin OP)
- retour du bras (commande T- du vérin T)

L'opération de matriçage-évacuation en comprend quatre:

- matriçage de la pièce (commande du vérin M)
- retour du vérin M (retour par ressort de rappel du vérin M)
- évacuation de la pièce par entrée du vérin E (commande du vérin E)
- retour à la position initiale du vérin E (retour par ressort de rappel du vérin E)

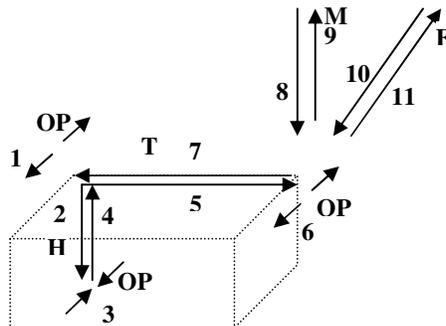


Figure 3.3 Cycle de fonctionnement du procédé AME.

III.2. Synthèse du superviseur

III.2.a. Recherche des états interdits

III.2.a.1. Grafquets modélisant le fonctionnement du procédé AME

Les spécifications de fonctionnement donnent d'une part les conditions initiales de démarrage du cycle de chaque sous système, et d'autre part les contraintes de synchronisation des deux sous systèmes.

Les conditions initiales du sous système 1 sont : $CI_1 = \text{Auto} \cdot \overline{AU} \cdot t_0 \cdot h_1$. Elles sont visualisées par le voyant "Etat initial" V2. Celles du sous système 2 ($CI_2 = m_0 \cdot e_0$) sont visualisées par le voyant "Etat initial" V3 (la signification des noms de variables est indiquée dans le tableau 3 des adresses des entrées/sorties).

Les contraintes de synchronisation sont les suivantes :

- le matriçage ne peut s'effectuer qu'après la détection de l'information de fin de d'alimentation (retour du vérin de translation),
- l'alimentation en pièces ne peut avoir lieu qu'après la détection de l'information de fin d'éjection (retour du vérin d'éjection).

La figure 2 donne les *Grafquets* des deux sous systèmes et celui de la synchronisation.

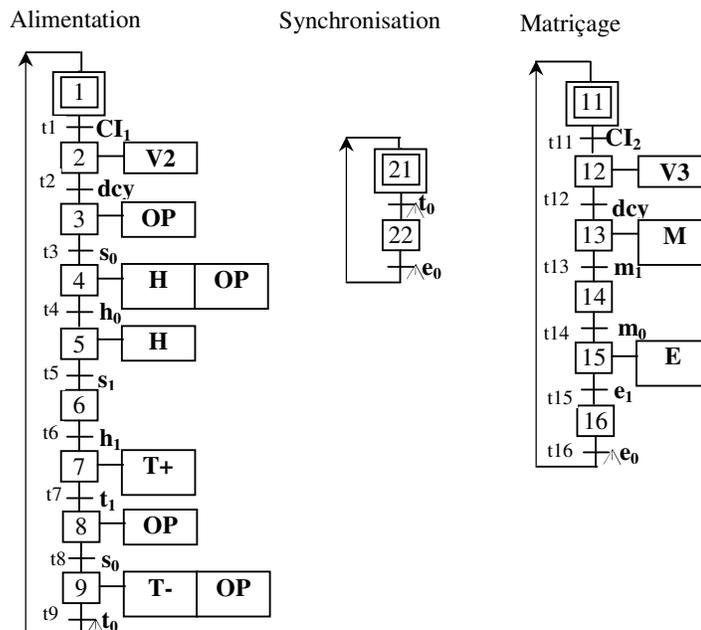


Figure 3.4 Grafquets partiels du procédé AME.

La synchronisation se fait grâce aux événements $\uparrow t_0$ et $\uparrow e_0$, l'information $\uparrow t_0$ indique la fin de la translation pour permettre le commencement du matriçage. L'information $\uparrow e_0$ indique la fin de l'éjection (retour du vérin) pour permettre l'alimentation du poste par une nouvelle pièce. Le principe de base est que chaque événement de synchronisation doit donner l'ordre d'action pour aboutir à l'événement complémentaire dont la synchronisation ($\uparrow t_0$ et $\uparrow e_0$ sont complémentaire).

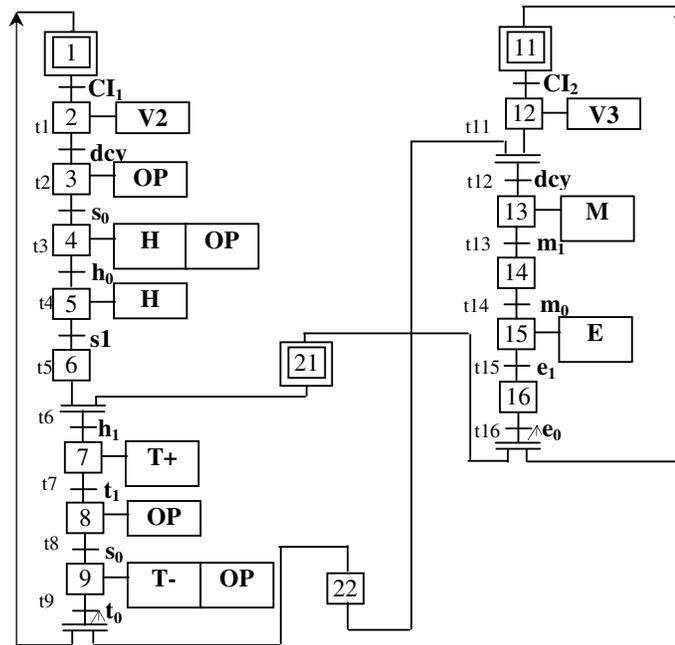


Figure 3.5 Grafcet de commande global de AME.

La commande des vérins simple et double effet provoque des événements contrôlables. Le retour par ressort de rappel (absence de commande) des vérins simple effet provoque des événements incontrôlables.

Par commodité, on identifie les transitions aux événements qui permettent de les franchir.

Les transitions à événements contrôlables sont

$$t1=CI_1; t2=t12=dcy; t3= t8=s_0; t4=h_0; t7=t_1; t9=\uparrow t_0; t11=CI_2; t13=m_1; t15=e_1$$

Les transitions à événements incontrôlables sont

$$t5=s_1; t6=h_1; t14=m_0; t16=\uparrow e_0$$

III.2.a.2. Graphe d'états accessibles du procédé AME

Le graphe d'états accessibles décrit les phases successives par lesquelles passera le système.

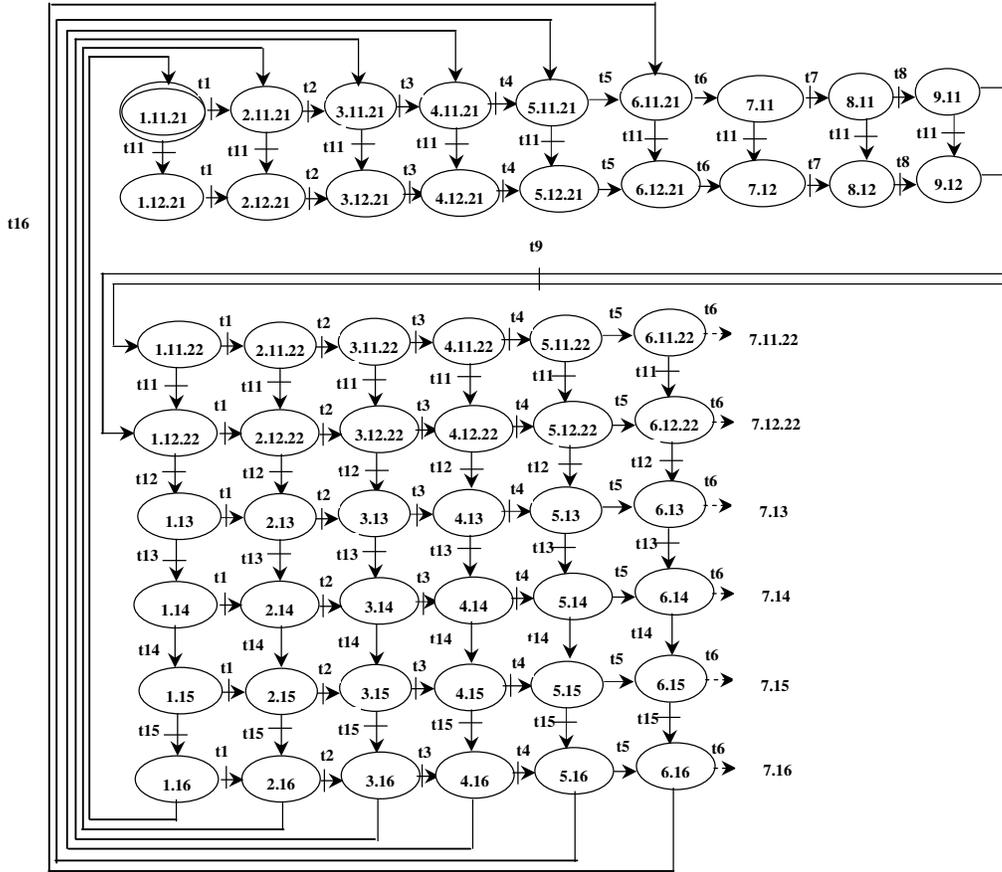


Figure 3.6 Graphe d'état accessible du procédé AME non supervisé.

III.2.a.3. Extraction des anomalies

Commençons par définir les états frontières E_B , interdits E_I et admissibles E_A .

D'après le graphe d'états accessibles:

$$E_A = \{ E_1E_{11}E_{21}, E_2E_{11}E_{21}, E_3E_{11}E_{21}, E_4E_{11}E_{21}, E_5E_{11}E_{21}, E_6E_{11}E_{21}, E_7E_{11}, E_8E_{11}, E_9E_{11}, E_1E_{12}E_{21}, E_2E_{12}E_{21}, E_3E_{12}E_{21}, E_4E_{12}E_{21}, E_5E_{12}E_{21}, E_6E_{12}E_{21}, E_7E_{11}, E_8E_{11}, E_9E_{11}, E_1E_{11}E_{22}, E_2E_{11}E_{22}, E_3E_{11}E_{22}, E_4E_{11}E_{22}, E_1E_{12}E_{22}, E_2E_{12}E_{22}, E_3E_{12}E_{22}, E_4E_{12}E_{22}, E_1E_{13}, E_2E_{13}, E_3E_{13}, E_4E_{13}, E_1E_{14}, E_2E_{14}, E_3E_{14}, E_4E_{14}, E_1E_{15}, E_2E_{15}, E_3E_{15}, E_4E_{15}, E_1E_{16}, E_2E_{16}, E_3E_{16}, E_4E_{16} \}. \tag{3.1}$$

$$E_I = \{ E_7E_{11}E_{22}, E_7E_{12}E_{22}, E_7E_{13}, E_7E_{14}, E_7E_{15}, E_7E_{16} \}. \tag{3.2}$$

$$E_B = \{ E_5E_{11}E_{22}, E_5E_{12}E_{22}, E_5E_{13}, E_5E_{14}, E_5E_{15}, E_5E_{16} \}. \tag{3.3}$$

III.2.b. Détermination du superviseur

III.2.b.1. Réduction des contraintes linéaires

- On exprime les états frontières de l'exemple sous forme de contraintes linéaires, on obtient :

$$\left. \begin{aligned} E_5 + E_{11} + E_{22} &\leq 2 \\ E_5 + E_{12} + E_{22} &\leq 2 \\ E_5 + E_{13} &\leq 1 \\ E_5 + E_{14} &\leq 1 \\ E_5 + E_{15} &\leq 1 \\ E_5 + E_{16} &\leq 1 \end{aligned} \right\} \quad (3.4)$$

- La construction des ensembles de *sur états* \mathcal{B}_1 et \mathcal{A}_1 pour les états frontières et admissibles est donné par :

$$\mathcal{B}_1 = \{ E_5, E_{11}, E_{12}, E_{13}, E_{14}, E_{15}, E_{16}, E_{22}, E_5E_{11}, E_5E_{22}, E_{11}E_{22}, E_5E_{12}, E_{12}E_{22}, E_5E_{13}, E_5E_{14}, E_5E_{15}, E_5E_{16}, E_5E_{11}E_{22}, E_5E_{12}E_{22} \}. \quad (3.5)$$

$$\mathcal{A}_1 = \{ E_1, E_2, E_3, E_4, E_5, E_6, E_7, E_8, E_9, E_{11}, E_{12}, E_{13}, E_{14}, E_{15}, E_{16}, E_{21}, E_{22}, E_1E_{11}, E_2E_{11}, E_3E_{11}, E_4E_{11}, E_5E_{11}, E_6E_{11}, E_7E_{11}, E_8E_{11}, E_9E_{11}, E_{22}E_{11}, E_{21}E_{11}, E_1E_{12}, E_2E_{12}, E_3E_{12}, E_4E_{12}, E_5E_{12}, E_6E_{12}, E_7E_{12}, E_8E_{12}, E_9E_{12}, E_{22}E_{12}, E_{21}E_{12}, E_1E_{21}, E_2E_{21}, E_3E_{21}, E_4E_{21}, E_5E_{21}, E_6E_{21}, E_1E_{22}, E_2E_{22}, E_3E_{22}, E_4E_{22}, E_1E_{11}E_{21}, E_2E_{11}E_{21}, E_3E_{11}E_{21}, E_4E_{11}E_{21}, E_5E_{11}E_{21}, E_6E_{11}E_{21}, E_1E_{11}E_{22}, E_2E_{11}E_{22}, E_3E_{11}E_{22}, E_4E_{11}E_{22}, E_1E_{12}E_{21}, E_2E_{12}E_{21}, E_3E_{12}E_{21}, E_4E_{12}E_{21}, E_5E_{12}E_{21}, E_6E_{12}E_{21}, E_1E_{12}E_{22}, E_2E_{12}E_{22}, E_3E_{12}E_{22}, E_4E_{12}E_{22}, E_1E_{13}, E_1E_{14}, E_1E_{15}, E_1E_{16}, E_2E_{13}, E_2E_{14}, E_2E_{15}, E_2E_{16}, E_3E_{13}, E_3E_{14}, E_3E_{15}, E_3E_{16}, E_4E_{13}, E_4E_{14}, E_4E_{15}, E_4E_{16} \} \quad (3.6)$$

- La suppression des *sur états* admissibles de l'ensemble des *sur états* frontières donne :

$$\mathcal{B}_2 = \{ E_5E_{22}, E_5E_{13}, E_5E_{14}, E_5E_{15}, E_5E_{16}, E_5E_{11}E_{22}, E_5E_{12}E_{22} \}. \quad (3.7)$$

- La suppression des *sur états* frontière non minimaux (exprimés par d'autres *sur états* de cardinal plus petit) donne l'ensemble final des *sur états* frontières :

$$\mathcal{B}_2 = \{ E_5E_{22}, E_5E_{13}, E_5E_{14}, E_5E_{15}, E_5E_{16} \}. \quad (3.8)$$

- Le choix des *sur états* qui couvrent un maximum de contraintes linéaires est effectué, sur la table de réduction 3.1, par la méthode de Mc Cluskey.

Tableau 3.1 Table de réduction sur états – contraintes.

$\mathcal{B}_2 \setminus E_B$	$E_5E_{11}E_{22}$	$E_5E_{12}E_{22}$	E_5E_{13}	E_5E_{14}	E_5E_{15}	E_5E_{16}	
E_5E_{22}	/	/					✓
E_5E_{13}			/				✓
E_5E_{14}				/			✓
E_5E_{15}					/		✓
E_5E_{16}						/	✓

L'ensemble réduit de *sur états* obtenus est donné par :

$$C_2 = \{ E_5 E_{22}, E_5 E_{13}, E_5 E_{14}, E_5 E_{15}, E_5 E_{16} \}. \quad (3.9)$$

- Fusionnement de contraintes par recherche d'éléments communs

L'expression C_2 indique qu'on a un seul ensemble de contraintes linéaires. Exprimée sous la forme donnée par (24), l'étape E_5 apparaît comme élément commun E_{CM} pour cet ensemble de contraintes.

$$E_5 + E_{22} \leq 1, E_5 + E_{13} \leq 1, E_5 + E_{14} \leq 1, E_5 + E_{15} \leq 1, E_5 + E_{16} \leq 1. \quad (3.10)$$

$$\{ E_{22} E_{13}, E_{22} E_{14}, E_{22} E_{15}, E_{22} E_{16}, E_{13} E_{14}, E_{13} E_{15}, E_{13} E_{16}, E_{14} E_{15}, E_{14} E_{16}, E_{15} E_{16} \} \notin A_1 \quad (3.11)$$

Comme aucun élément de l'ensemble donné par l'expression (25) n'appartient à \mathcal{A}_1 , on peut alors remplacer les contraintes de l'expression (24) par celles de (26).

$$E_5 + E_{22} + E_{13} + E_{14} + E_{15} + E_{16} \leq 1. \quad (3.12)$$

L'ensemble C_1 est alors donné par $C_1 = \{ E_5 E_{22} E_{13} E_{14} E_{15} E_{16} \}. \quad (3.13)$

III.2.b.2. Construction du superviseur par l'application du principe de précedence

Le graphe d'états accessibles simplifié (*GEAS*) est obtenu par la suppression des états interdit et frontière du graphe d'état accessible.

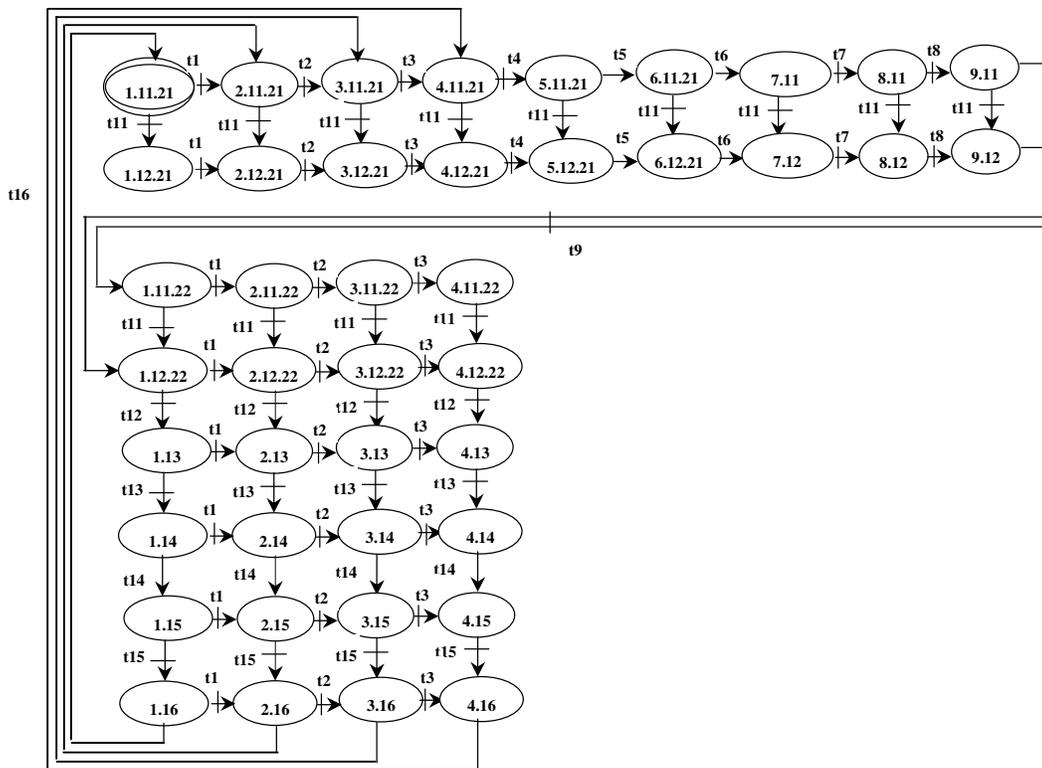


Figure 3.7 Graphe d'états accessibles simplifié du procédé AME non supervisé.

- **Recherche de $E_{Pré}$ et E_{Suc} pour l'étape commune E_{CM} .**

On localise sur le *GEAS* les occurrences de E_{22} E_{13} E_{14} E_{15} E_{16} par rapport à E_5 pour déterminer les $E_{Pré}$ et E_{Suc} .

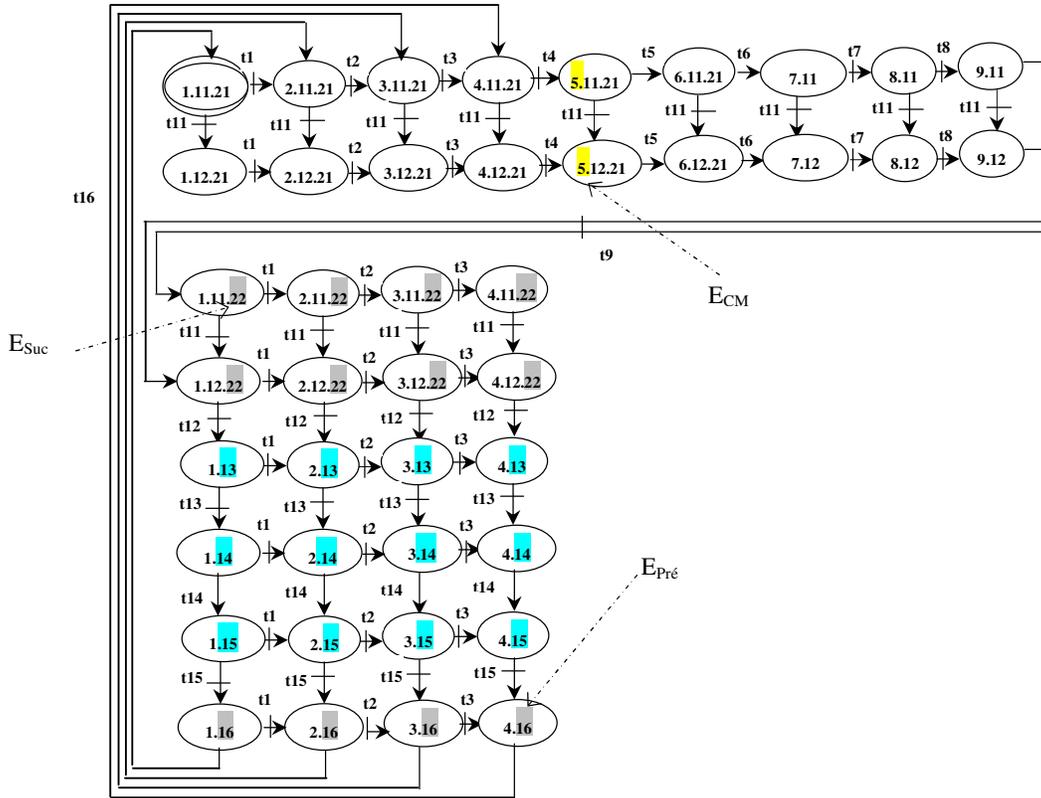


Figure 3.8 Le *GEAS* avec ordre d'occurrence de $E_{Pré}$ et E_{Suc} .

En parcourant le *GEAS* en mode avant puis arrière, on obtient que l'étape qui précède E_5 est E_{16} et celle qui lui succède est E_{22} . Par conséquent l'ordre d'occurrence est $E_{16} E_5 E_{22}$.

- **Vérification des conditions de précédence sur le graphe d'états accessibles**

Pour le groupe d'étapes E_{16} E_5 E_{22} on vérifie sur le *GEA* si depuis un état admissible qui inclut l'étape 16 on peut passer à un état frontière qui inclut l'étape 5, et si depuis un état admissible qui inclut l'étape 5 on peut passer à un état frontière qui inclut l'étape 22.

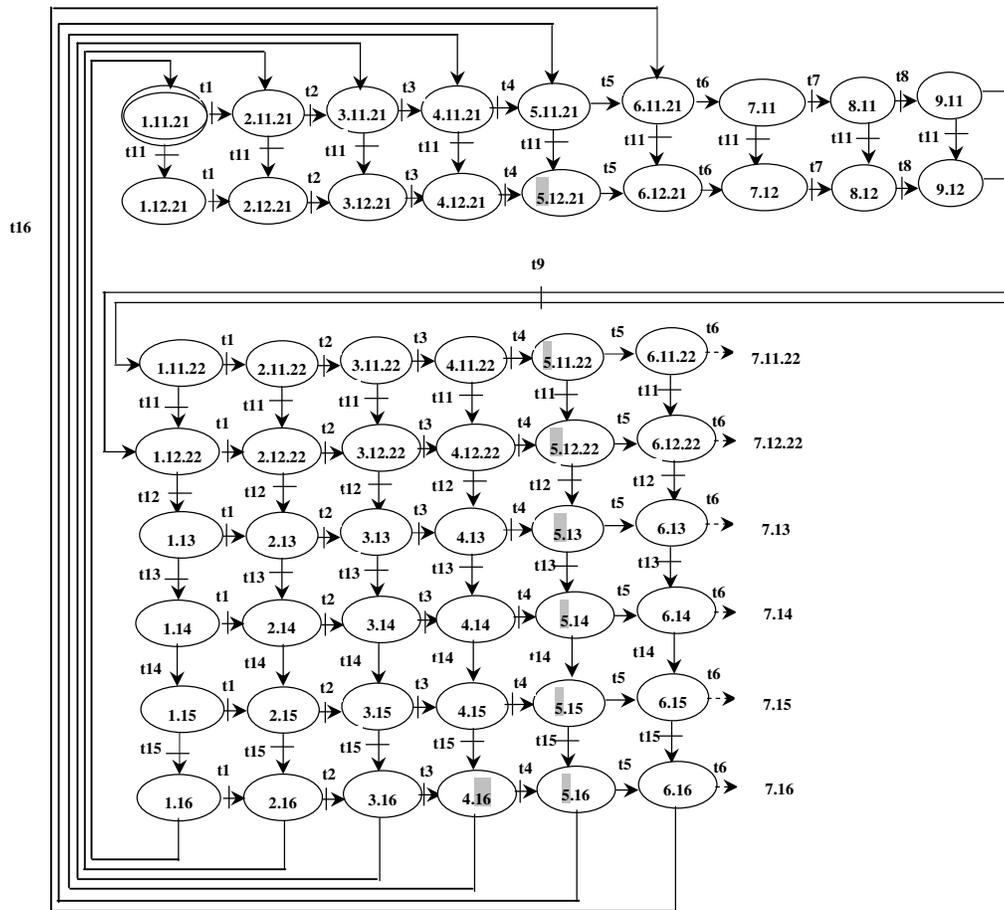


Figure 3.9 GEA avec les conditions de précédence.

On a une seule condition de premier type vérifiée, entre l'étape 16 et 5. On a donc une étape de contrôle E_{165} .

III.2.c. Fonctionnement du procédé supervisé

III.2.c.1. Introduction des étapes de supervision dans la commande

L'unique étape de contrôle E_{165} aura deux arcs : un en entrée de E_{16} vers E_{165} et un en sortie de E_{165} et vers E_5 .

Comme l'étape E_{CM} est unique, l'étape de contrôle E_{165} sera initialement active.

Le superviseur qui permet de satisfaire les contraintes linéaires données par l'expression (18) est exprimé par l'étape de contrôle E_{165} .

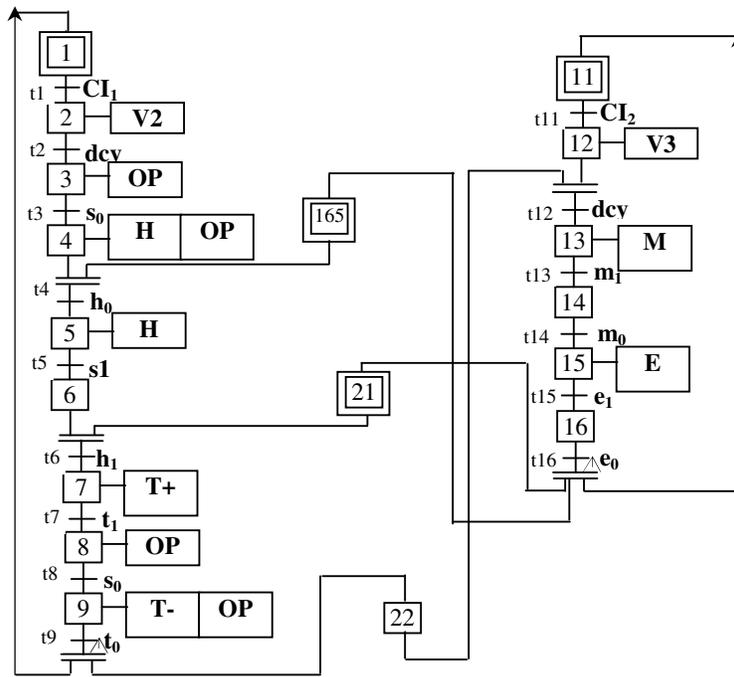


Figure 3.10 Grafcet de commande supervisé de AME.

III.2.c.2 Vérification du fonctionnement du système en boucle fermée

Construisons le graphe d'états accessibles du *Grafcet* de commande supervisé du procédé.

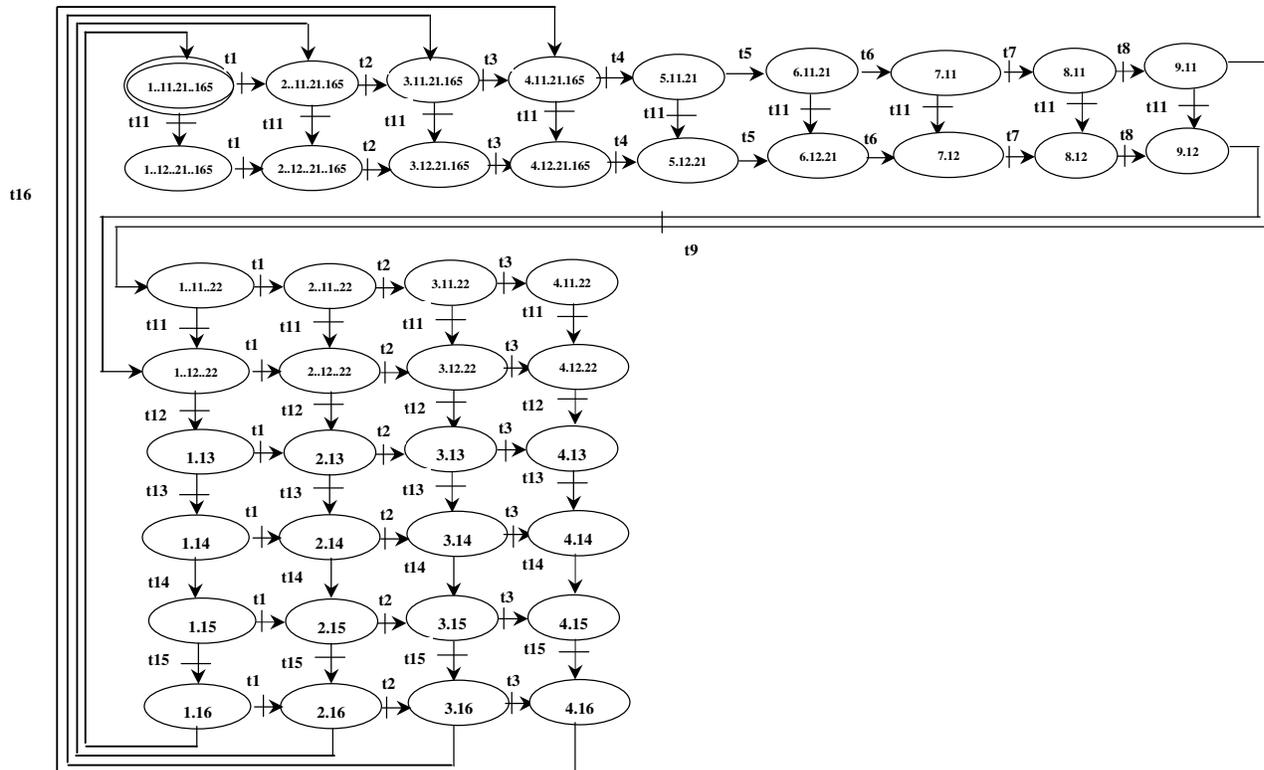


Figure 3.11 Graphe d'états accessibles sous superviseur de AME.

On remarque que le graphe d'états accessibles avec superviseur n'est constitué que d'états admissibles. L'étape de contrôle E_{165} a permis de satisfaire les équations obtenues depuis les contraintes linéaires, en interdisant l'occurrence des états frontières et interdits.

L'étape de contrôle E_{165} est activée en même temps que l'étape de synchronisation E_{21} , suite au franchissement de t_{16} . D'autre part E_{165} réagit sur le *Grafcet* de commande en amont de E_{21} , cela assure que la tâche qui doit être accomplie par l'étape de synchronisation est prise en charge par l'étape de contrôle. Donc l'étape E_{21} peut être supprimée sans aucune influence sur le fonctionnement.

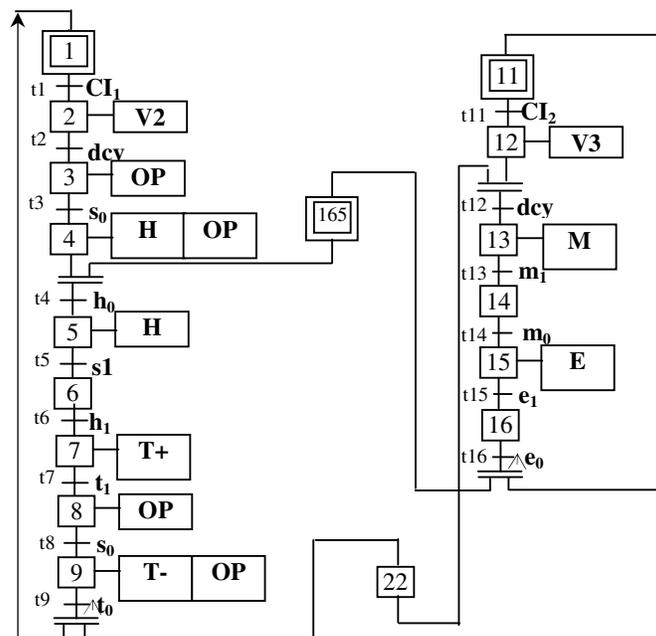


Figure 3.12 Grafcet de commande simplifié du procédé AME supervisé.

On remarque que les étapes de contrôle du superviseur sont utilisées pour éviter la défaillance due au franchissement des transitions incontrôlables liées à la synchronisation.

Dans notre cas seule l'étape de synchronisation E_{21} aboutit à une transition incontrôlable. Elle sera remplacée par la supervision. L'autre étape de synchronisation E_{22} aboutit à une transition contrôlable. Par conséquent elle n'est pas concernée par la supervision et ne peut être supprimée.

III.3. Implémentation du Grafcet de commande supervisé

Depuis le *Grafcet* de commande supervisé de la figure 3.12, on réalise les programmes séquentiel et postérieur en langage PL7-2, pour l'automate TSX 17-20.

III.3.a. Adressage des entrées / sorties

Les tables 3.2 et 3.3 donnent la correspondance entre les variables d'entrée / sortie et leurs adresses respectives.

	Nom variable	Commentaire	Adresse automate
Module de Base	R/S	interrupteur run/stop	I0.00
	t ₁	pince en avant	I0.01
	t ₀	pince en arrière	I0.02
	m ₁	poinçon en bas	I0.05
	m ₀	poinçon en haut	I0.06
	e ₁	pièce évacuée	I0.07
	e ₀	évacuateur au repos	I0.08
	s ₁	pince fermée	I0.09
	s ₀	pince ouverte	I0.10
Module d'extension	h ₁	pince en haut	I1.00
	h ₀	pince en bas	I1.01
	Auto	interrupteur en position auto	I1.06
	Main	interrupteur en position main	I1.07
	Press	mise en pression de la PO	I1.08
	Dcy	départ cycle	I1.09
	Arret	arrêt	I1.10
	$\overline{\text{AU}}$	arrêt d'urgence	I1.11
	$\overline{\text{Valid}}$	validation redémarrage après détection de AU	I1.12

Table 3.2 Adresses des entrées.

	Nom variable	Commentaire	Adresse automate
Module de Base	A	alimenter le signal de référence des bloqueurs, capteurs pneumatiques, et du sectionneur	O0.00
	T+	translater avant	O0.01
	T-	translater arrière	O0.02
	M	descente poinçon	O0.05
	E	évacuer pièce	O0.06
	OP	ouvrir pince	O0.07
Module d'extension	H	descente pince	O1.01
	V1	Allumer voyant " <i>sous pression</i> "	O1.06
	V2	allumer voyant " <i>état initial</i> " sous Système 1	O1.07
	V3	allumer voyant " <i>état initial</i> " sous système 2	O1.09

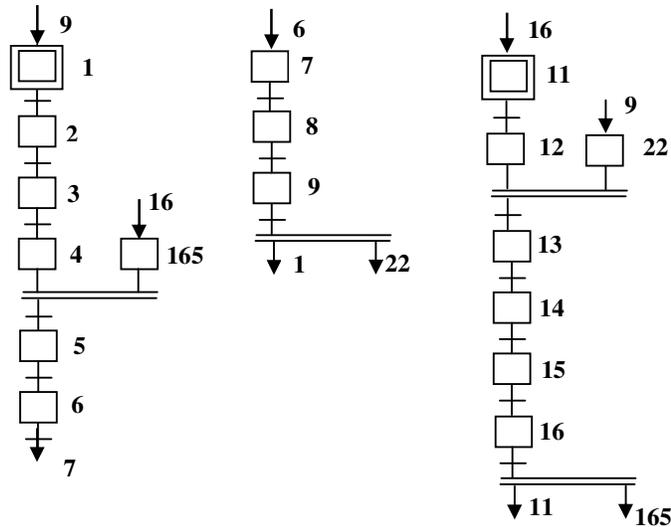
Table 3.3 Adresses des sorties.

III.3.b. Programme séquentiel en langage PL7-2

• **Grafcet de séquençement étape_transition**

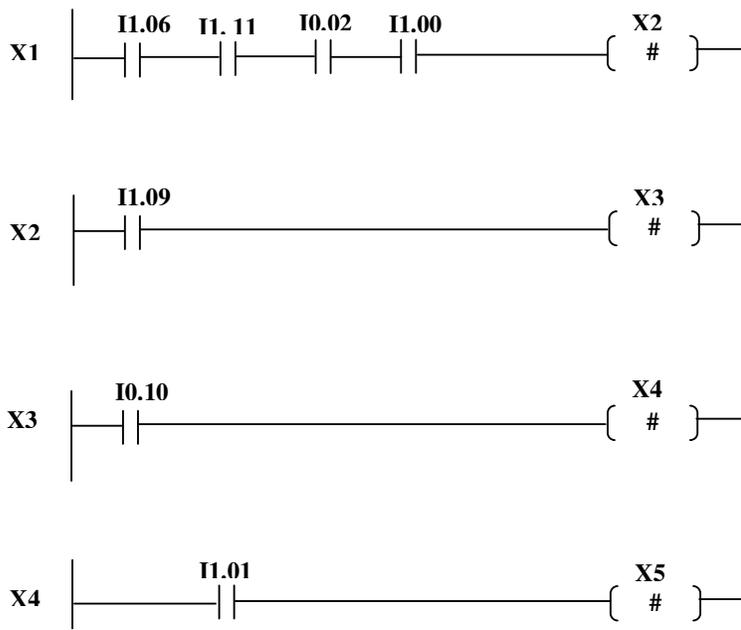
Ce graphe est dessiné grâce aux primitives graphiques : étape, transition, flèche d'origine et de destination, double barre de jonction de synchronisation : entrée ou sortie de séquences simultanées.

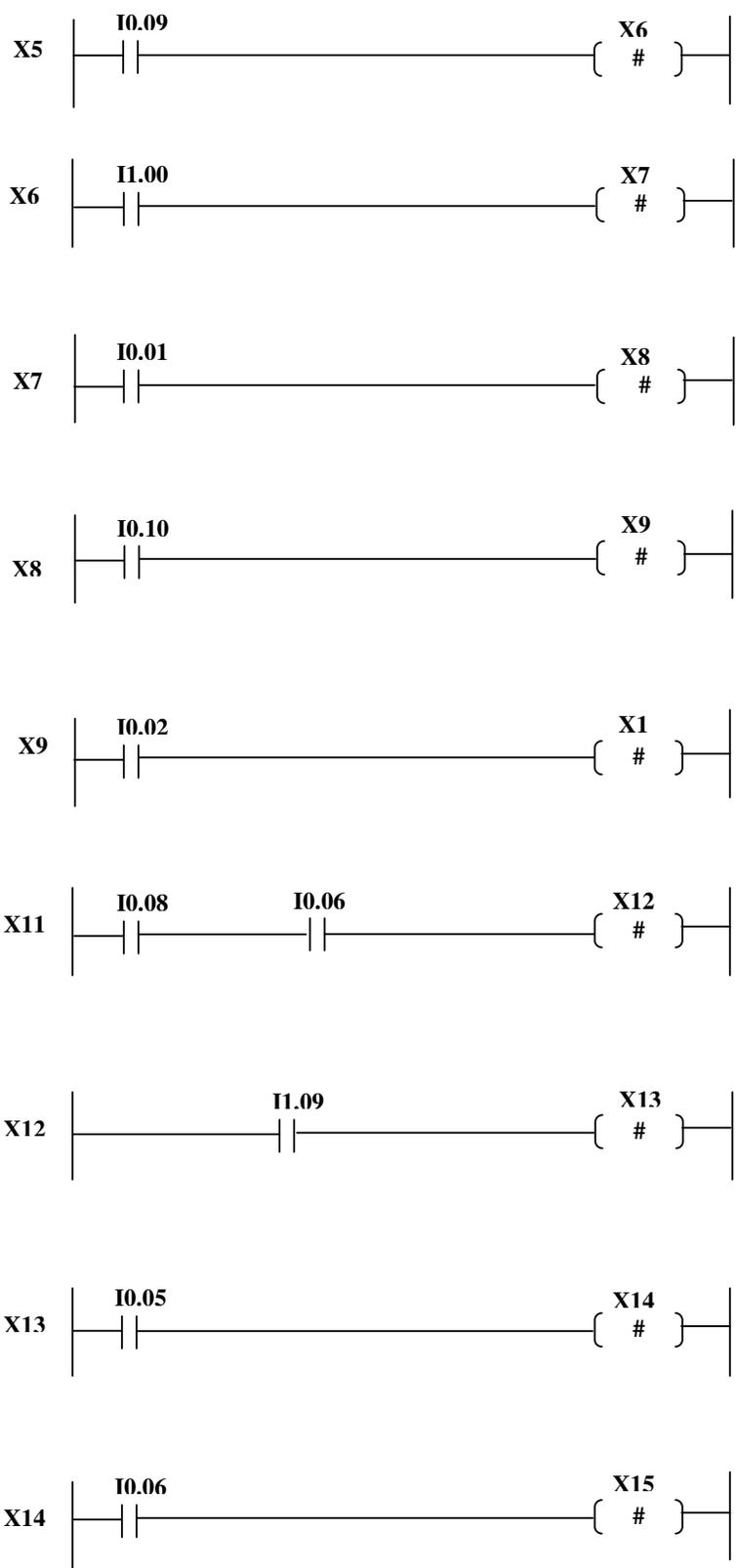
P0



• **Entrées ou Réceptivités associées aux transitions (en Langage LADDER)**

Pour chaque transition du *Grafcet*, on a un réseau de contacts étiqueté par l'étape précédant la transition. En sortie (action) du réseau, on a les informations de la réceptivité permettant le franchissement de la transition.

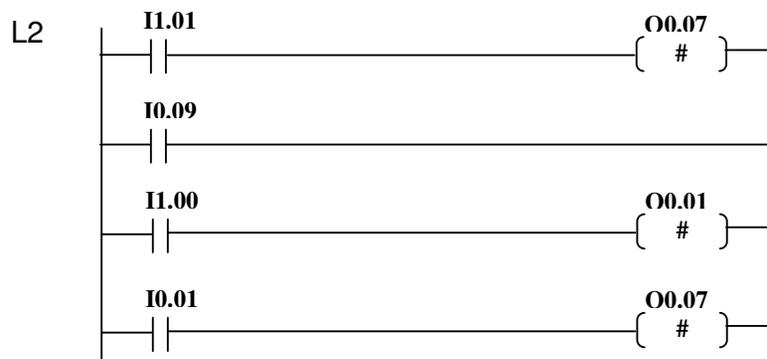
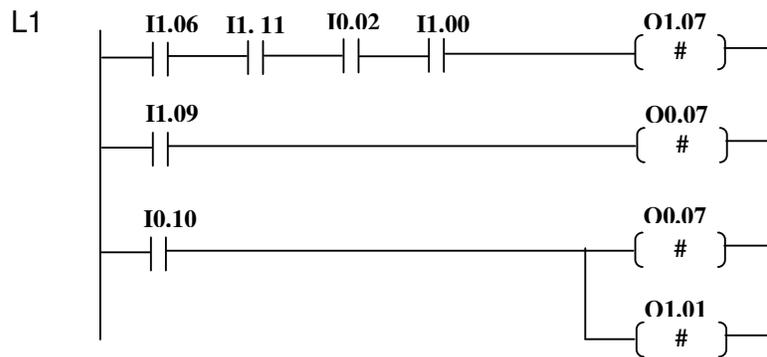


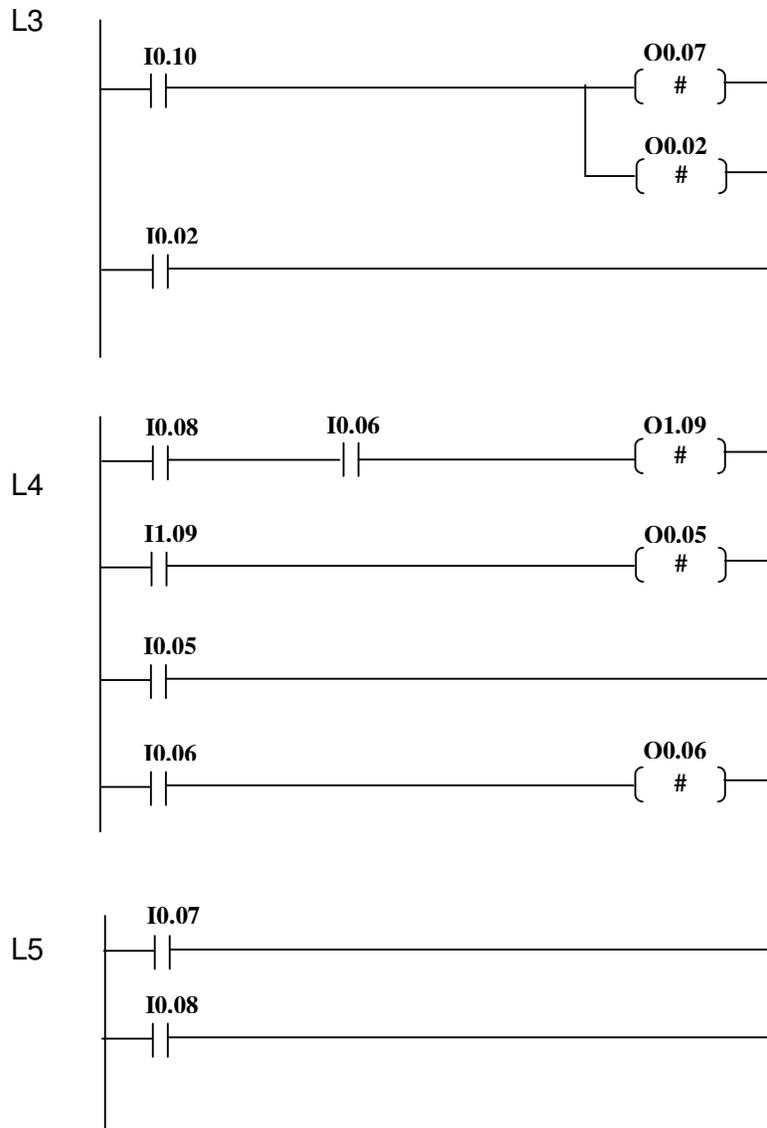




III.3.c. Sorties ou programme postérieur (actions) en langage LADDER

L'activation des actions du *Grafcet* est faite par le biais de réceptivités, ce qui est exprimé par des groupes de réseau étiqueté (Li) dont la taille est limitée à quatre lignes.





III.3.d. Programme préliminaire

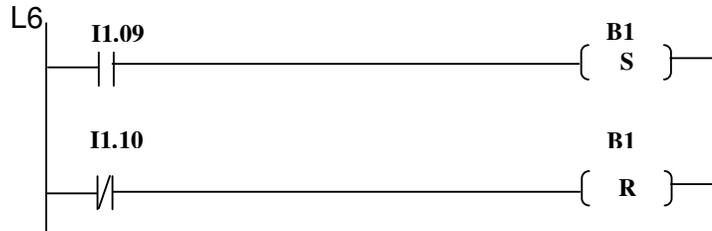
III.3.d.1. Fonctionnement automatique

Notre *Grafset* traduit un fonctionnement cycle par cycle. Un nouveau cycle ne se produit qu'après chaque nouvel appui par l'opérateur sur le bouton poussoir dcy.

La commande supervisée trouve son intérêt dans un fonctionnement automatique, où l'opérateur humain n'intervient que pour donner l'ordre de départ cycle par appui sur dcy, et l'ordre d'arrêt par appui sur un bouton poussoir "Arrêt".

Ce type de fonctionnement est possible en rajoutant le programme préliminaire ci-après.

La variable interne B1 est de type mémorisé. Elle est mise à 1 par l'appui sur dcy, et remise à zéro par la variable Arrêt.



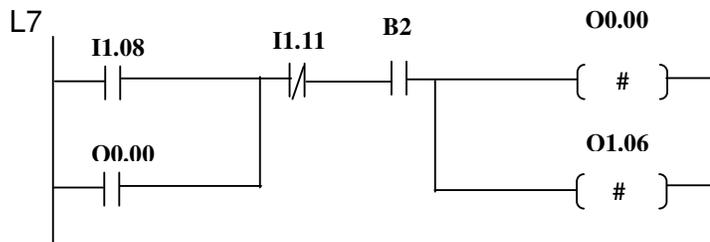
Ainsi sur le *Grafcet* de la figure 3.10, la variable dcy sera remplacée par le bit interne B1, et sur le réseau de contacts étiqueté par X2 (passage de l'étape 2 à l'étape3), l'adresse I1.09 sera remplacée par B1.

III.3.d.2. Mise sous pression et sécurité de fonctionnement

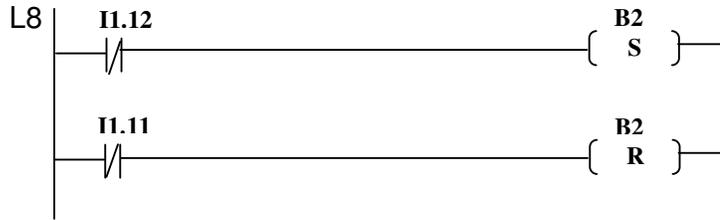
La mise sous pression de l'installation se fait séparément et n'apparaît pas sur le *Grafcet*. Elle est prise en charge par un programme séparé en langage à contacts.

Par appui sur le bouton poussoir « Press » (mise sous pression), le voyant V1 sera allumé et on envoie le signal d'alimentation en pression A du sectionneur général.

Par sécurité, l'appui sur l'arrêt d'urgence provoque une coupure de l'alimentation. La disparition du signal d'urgence et le réenclenchement de la mise en pression ne suffisent pas pour rétablir l'alimentation. Un bouton poussoir de validation de la suppression de l'urgence permet de réaliser un deuxième niveau de sécurité (ce bouton est sous clef sous la responsabilité du chef d'atelier).



Le bouton de validation est initialement actif. La présence de l'arrêt d'urgence provoque sa désactivation.



III.4.Conclusion

La technique de supervision développée nous a permis d'obtenir un superviseur optimal. Cet exemple a permis de mettre en relief le fait que l'étape de synchronisation qui agit sur l'événement contrôlable n'a pas été supprimée. Par contre celle qui agit sur celui incontrôlable a bien été supprimée. Le superviseur obtenu se compose d'une étape de contrôle qui remplace l'étape de synchronisation et assure l'élimination des défaillances.

L'implémentation de la commande supervisée sur l'API (TSX17-20) s'est faite directement, grâce à l'outil de modélisation utilisé (**Grafcet**) qui est en même temps une partie du langage de programmation de l'API (*PL7-2*). Les étapes et les transitions du Grafcet de commande supervisé représentent le Grafcet de séquençement. L'ensemble des réceptivités associés aux transitions sont exprimées en langage *ladder*, et donnent le programme séquentiel, Pour avoir le programme postérieur il suffit d'exprimer en langage *ladder* les conditions nécessaires à chaque action.

Le programme préliminaire prend en charge la mise sous pression de l'installation pneumatique et assure un cycle de travail automatique sécurisé.

CONCLUSION

CONCLUSION

Le but de la synthèse du superviseur des systèmes à événement discret est de satisfaire les spécifications de la commande. Plusieurs techniques et outils de modélisation ont été développés afin d'obtenir un superviseur simple avec une implémentation directe, ce qui représente un défi.

La majorité des études qui ont abouti à un superviseur simple ont été confrontées au problème de l'implémentation, c'est le cas notamment des études basées sur le *RDP* ([8] et [11]). Par contre celles qui ont utilisé le *Grafcet* (ce qui facilite l'implémentation) ont échoué dans la simplification ([7]). D'autres ont utilisé les résultats obtenus par le *RDP* pour les appliquer au *Grafcet* ([12]), mais cela a été beaucoup plus une traduction du *RDP* au *Grafcet*, avec tous les inconvénients que cela comporte, car les spécifications des deux outils ne sont pas les mêmes.

Dans notre étude nous avons pu obtenir un superviseur simple avec implémentation directe, car l'outil *Grafcet* a été utilisé de bout en bout, de la modélisation à l'implémentation.

Dans la synthèse du superviseur nous avons conservé les définitions de base des SED concernant les états frontières, admissibles et interdits, mais nous avons adapté au *Grafcet* certains résultats obtenus à partir des techniques de supervision basées sur le *RDP*, sans être pris par le piège de la traduction. Ainsi nous avons développé une technique de supervision différente.

Pour les transitions incontrôlables provoquant un nombre important de contraintes linéaires, nous avons utilisé la méthode de *sur état* pour les réduire.

Nous avons développé également une deuxième méthode de réduction, qui est basée sur le principe de précedence des étapes de la contrainte. Cette réduction de contraintes a permis d'obtenir un nombre réduit de places de contrôle, avec un nombre minimal d'arcs.

La comparaison des résultats avec ceux obtenus par la technique de *Kattan* nous a permis de confirmer que notre méthode aboutit à un superviseur de taille minimale.

Par ailleurs nous avons pu simplifier le *Grafcet* supervisé par élimination des étapes de synchronisation, dont le fonctionnement est couvert par le superviseur. Ce qui offre une représentation plus lisible.

Le deuxième aspect concerne l'implémentation du superviseur. L'utilisation du *Grafcet* comme outil pour modéliser la commande et la supervision a permis une

implémentation aisée sur l'automate programmable industriel TSX17-20 de télémécanique. De plus cela s'effectue de manière systématique, car il suffit presque d'écrire le *Grafset* de commande et de supervision dans le langage de l'automate (PL7.2). Nous avons ajouté d'autres programmes, non prévus initialement dans le cahier de charges, pour assurer un fonctionnement automatique et sécurisé du système.

Nous pouvons dire que les objectifs de supervision sont accomplis grâce à la technique de simplification et le bon choix de l'outil de modélisation. Cela ouvre la voie à d'autres études, comme:

- systématiser les étapes de synthèse du contrôleur par le développement d'un logiciel spécifique,
- supprimer les étapes de synchronisation de façon systématique et non pas intuitive,
- tester la technique de supervision développée sur des systèmes plus compliqués,
- appliquer les résultats obtenus au cas des systèmes temporisés (prise en considération de l'aspect temporel dans le franchissement des transitions).

Références bibliographiques

[1] G.Scorletti, G.Binet, «Réseaux de Pétri », cours EL401T2, master 1A mention EEA Parcours AEII, université de Caen/Basse Normandie, pp22-24, Juin 2006.

[2] J-L.FERRIER, J-L. BOIMOND, « Systèmes dynamiques à événements discrets du modèle à la commande», monographie, ISTIA - université d'Angers, Laboratoire d'Ingénierie des Systèmes Automatisés (LISA), Juin 2004, <http://www.istia.univ-angers.fr/LISA>.

[3] J.P.Chemla, «Modélisation et analyse des systèmes de production», cours - département productive, école polytechnique tours, France 2000.

[4] J.M.Roussel, « Analyse de Grafsets par génération logique de l'automate équivalent», thèse de doctorat de l'Ecole Normale Supérieure de Cachan, France 1994.

[5] Z.Achour, A.Ghaffari, N.Rezg, X.Xie, «Petri net controller synthesis using theory of regions », 1st IPROMS Virtual International Conference on Intelligent Production Machines and Systems, IPROMS 2005, 4-15 Juillet 2005. http://conference.iproms.org/iproms_2005_conference

[6] P.J.Ramadge, W.M.Wonham, «On the supremal controllable sublanguage of a given language», SIAM journal de contrôle et optimisation, 25, pp 637-659, 1987.

[7] F.Charbonnier, « Commande supervisée des systèmes à événement discret », Thèse de doctorat, INPG France, 1996.

[8] K.Yamalidou, J.Moody ,M.Lemmon, P.Antsaklis, « Feed back control Pétri net based on place invariant », Automatica, Pergamon, (32)1, pp15-28, 1996.

[9] J.O.Moody, P.Antsaklis, «Petri net supervisors for DES with uncontrollable and unobservable transition», IEEE Trans. on Automatic Control, 45(3), pp 462-476, Mars 2000.

[10] A.Dideban et H.Alla, « Solving the problem of forbidden states by feedback control logical synthesis », The 32nd Annual Conference of the IEEE Industrial Electronics Society, Paris-France 7-10 Nov 2006.

[11] A.Dideban, « Synthèse de contrôleurs discrets par simplification de contraintes et de conditions », thèse de doctorat, Université Joseph Fourier-Grenoble, France 2007.

[12] B.Kattan , «Synthèse structurelle d'un contrôleur basée sur le grafcet », thèse de doctorat, Université Joseph Fourier-Grenoble, France - 2004.

[13] R.KUMAR, «Supervisory synthesis techniques for discrete event dynamical systems», thèse de doctorat, Université de Texas, Austin, Août 1991.

[14] M.Mano, «Digital design», Prentice hall, 2001.

[15] R.David, H.Alla, «Discrete, Continuous, and Hybrid Petri Nets», ch 2 pp34, Springer, 2005.

Résumé

Dans le contrôle supervisé des systèmes à événements discrets, les transitions incontrôlables sont la source d'états interdits. La majorité des études développées pour résoudre ce problème sont basées sur le RDP. On présente ici une approche pour la vérification des propriétés d'un Grafcet de commande (présentant des transitions de synchronisation), la suppression des états interdits, puis la validation par rapport au cahier des charges.

La technique de « précedence », développée pour synthétiser le superviseur, repose sur l'extraction de l'automate équivalent au Grafcet (graphe d'états accessibles). Elle permet d'obtenir le superviseur optimal : nombre minimal d'étapes et d'arcs.

Le Grafcet est utilisé de bout en bout : de la modélisation à l'implémentation sur API. Les résultats obtenus ont été testés sur un système industriel de matriçage et évacuation de pièces, puis implémentés sur un TSX 17-20 programmé en PL7-2.

Abstract

In the supervised control of the systems with discrete events, the uncontrollable transitions are the source of prohibited states. The majority of the studies developed to solve this problem are based on the RDP. we presents here an approach for the checking of the properties of Grafcet command (presenting transitions of synchronization), the removal of the prohibited states, then the validation compared to the schedule of conditions.

The technique of "precedence", developed in order to synthesize the supervisor, rests on the extraction of the automat equivalent of the Grafcet (graph of accessible states). It allows obtaining the optimal supervisor: minimal number of stages and arcs.

The Grafcet is used from beginning to end: to the modelling at the implementation on API. The results obtained were tested on an industrial system of dieing and evacuation of pieces, then implemented on a TSX 17-20 programmed in PL7-2.

ملخص

في التحكم المتنبأ للأنظمة ذات الحوادث المتقطعة, الحوادث غير المتحكم بها تمثل منبع الحالات المحظورة. معظم الدراسات التي تم تطويرها لحل هذا الأشكال اعتمدت على *RDP*. نقدم مقاربة لالتحقق من خواص *Grafcet* التحكم (تمثل وصلات توافق) حذف الحالات المحظورة ثم تصديقها بالنسبة لدفتر الشروط.

تقنية 'التنبأ', طورت من اجل تركيب وسيلة تنبأ تشكيل الآلية المكافئة ل *Grafcet* (تمثيل الحالات المحظورة) يسمح بالحصول على وسيلة تنبأ فضلى : اقل عدد من الوضعيات و الأقواس.

Grafcet تم استعماله من طرف لطرف : من التمثيل الى التركيب على *API*. النتائج المتحصل عليها اختبرت على نظام الصناعي لقولبة و إجلاء القطع ثم التركيب على TS X17-20 مبرمج ب PL7-2.