

## THÈSE

Pour obtenir le grade de

## DOCTEUR DE L'UNIVERSITÉ DE GRENOBLE

Spécialité : **Automatique-Productique**

Arrêté ministériel : 7 août 2006

Présentée par

**Sonia BATIS**

Thèse dirigée par **Hassane ALLA**

préparée au sein du **GIPSA-Lab** dans l'école doctorale **EEATS**

## Commande d'une classe de systèmes hybrides par automates hybrides rectangulaires

Thèse soutenue publiquement le **18 Septembre 2013**,  
devant le jury composé de :

**M. Jean-Marc THIRIET**

Professeur, Université Joseph Fourier, Président

**Mme. Isabel DEMONGODIN**

Professeur, Université U3, Rapporteur

**M. Jean-Jacques LESAGE**

Professeur, ENS Cachan, Rapporteur

**M. Kondo ADJALLAH**

Professeur, ENI Metz, Examineur

**M. Hassane ALLA**

Professeur, UJF Grenoble, Directeur de thèse





## Dédicaces

Le présent travail vient couronner plus d'une vingtaine d'années d'études pendant lesquelles j'ai été accompagnée et soutenue par ceux que j'aime de tout mon cœur:

Mes parents **Habib** et **Narjès** (Harrarou et Mamichou), sans qui je ne serais pas là aujourd'hui. Je tiens à leur dédier cette précieuse thèse pour leur témoigner mon amour éternel, mon admiration pour leur rayonnement aussi bien familial que professionnel. Je voudrais tant leur ressembler et qu'ils soient fiers de moi. J'espère qu'ils trouvent dans ce travail un peu de ce qu'ils m'ont appris et donné. Qu'ils soient sûrs que je ne dévierai jamais des principes qu'ils m'ont inculqués (rigueur, honnêteté, travail bien fait). Je leur renouvelle mon affection et toute ma tendresse. Papa, Maman, vous êtes mes idoles, vous êtes mes deux étoiles, vous êtes ma source d'inspiration. Malgré ces trois années loin de vous, je vous ai gardés près de mon cœur et votre amour m'a toujours encadrée et protégée. Je vous aime infiniment.

Mon fiancé **Omar** (Azizi el ghaly). Sans son amour et son soutien quotidien pendant ces trois années de thèse, je ne sais pas si j'aurais aussi bien tenu loin des miens. Qu'il soit assuré de mon amour éternel. Mon cœur, merci de m'avoir encouragée et d'avoir toujours été là pour moi. Je t'aime passionnément!

Ma sœur **Raya** (Sistounty), avec qui nous avons partagé bien de bons moments de complicité, d'affection et de petites chamailleries entre deux sœurs qui s'adorent. J'espère que nous serons toujours aussi proches et unies. Sistounty, vas au bout de tes rêves. Je t'aime très fort!

Je n'oublierai pas de citer ma grand-mère **Habiba** (Mamie), mon symbole du courage, de l'amour de la vie, du défi et de la tendresse. C'est elle qui m'a appris à marcher. J'espère qu'elle est fière de moi et qu'elle soit toujours aussi dynamique et pleine de vie. Mon affection pour toi Mamie est infinie.

Mes beaux parents **Béehir** et **Mabrouka**. Merci de votre soutien et de vos encouragements. Am Béehir, merci pour tes précieux conseils et pour les dattes que tu me choisis avec minutie. Tata Mabrouka, merci de ta gentillesse et pour tes si bons petits plats. Que Dieu vous garde et vous comble de joie.

Ma belle sœur **Rim**, symbole de courage et de douceur, tu es une vraie sœur pour moi et je te souhaite le bonheur et une excellente santé.

Mon beau-frère **Mourad**, sa femme **Mériem**, mon amie, et leur petite **Sarra** , par ces dédicaces, je voudrais vous témoigner mon affection et vous souhaiter le meilleur.

Mes oncles et mes tantes, vous avez attendu ce jour avec moi. J'espère avoir été digne de votre confiance. Une dédicace spéciale à mon oncle **Raouf**, champion des mots fléchés, et des blagues, à qui je souhaite une excellente santé et une très bonne forme. Et, une autre pour mon oncle **Wassil**, pour sa bonne humeur, son enthousiasme et ses bonnes pizzas( hi, hi, hi ...), sans oublier sa femme **Monia**, pour sa générosité. Je vous aime beaucoup et vous êtes tellement beaux ensemble !

La primeur à mes cousines et cousins avec qui j'ai joué, rigolé, et partagé plein de merveilleux moments. Je vous aime de tout mon cœur et vous souhaite le succès et le bonheur.

Si je suis, aujourd'hui là, c'est aussi grâce à mes maîtres et maîtresses, mes professeurs du Lycée pilote de Tunis, mes professeurs de la Faculté des Sciences de Tunis ainsi que ceux de l'Ecole Nationale des Ingénieurs de Tunis. Je vous dois tellement, et à travers cette thèse je vous rends un grand hommage et vous souhaite santé, bonheur et encore plus de rayonnement dans la lourde tâche de former pour notre **TUNISIE**, les générations futures.

Avec tout mon amour, **Sonia**

## Remerciements

Le présent travail a été réalisé au laboratoire Gipsa-lab de l'Institut National Polytechnique de Grenoble.

Mes vifs remerciements s'adressent à tous ceux qui m'ont aidée directement ou indirectement dans la préparation de ma thèse, et tout d'abord à Monsieur le Professeur **Hassane Alla** qui a, non seulement dirigé ce travail mais n'a épargné aucun effort pour me faire bénéficier d'une bourse pour l'accomplir. Je ne l'en remercierai jamais assez. Je tiens aussi, à lui témoigner mon admiration pour ses hautes qualités pédagogiques et humaines. Il m'a encouragée et épaulée, sans compter sa peine. J'ai toujours bénéficié de ses connaissances, de son expérience et de sa perspicacité pour mener à bien ce travail. J'espère avoir été à la hauteur de votre confiance, Monsieur le Professeur, et soyez assuré de ma très vive gratitude et de mon éternelle amitié. Je n'oublierai jamais votre soutien.

Je ne saurais exposer ce travail sans exprimer ma profonde reconnaissance à Monsieur le Professeur **Jean-Marc Thiriet**, qui m'a si gentiment accueillie dans le laboratoire et qui m'a accordé toutes les facilités pour mener à bien ce travail. Je suis tellement heureuse de l'honneur qu'il me fait en acceptant de présider le jury de ma thèse.

Il m'est aussi agréable de remercier Madame la Professeure **Isabel Demongodin** et Monsieur le Professeur **Jean-Jacques Lesage** pour avoir bien voulu se charger d'être les rapporteurs de ce travail et de faire partie de ce jury.

J'exprime également ma réelle gratitude à Monsieur le Professeur **Kondo Adjallah** pour l'honneur qu'il me fait d'avoir consenti à être l'examineur de ma thèse.

Un grand merci à Monsieur le Professeur **Christian Commault**, Directeur de l'Ecole Doctorale EEATS, pour son accueil et son aide précieuse. Il n'a jamais ménagé ses efforts pour être à l'écoute des besoins des doctorants.

Je voudrais aussi exprimer toute ma gratitude à Madame **Marie-Thérèse Descotes-Genon**, une femme exceptionnelle et généreuse. Je ne l'oublierai jamais. Elle m'a accueillie avec son beau sourire et m'a guidée dans toutes les démarches administratives au sein et en dehors du

laboratoire. Je voudrais te dire, **Marie-Thé**, que le labo n'est plus du tout le même depuis que tu es partie à la retraite. Merci de nous avoir traités comme tes enfants!

Je remercie aussi tous les membres du laboratoire Gipsa-lab, particulièrement Madame **Patricia Reynier**, notre bibliothécaire adorée, et tous les enseignants avec qui j'ai collaboré et auprès de qui, j'ai beaucoup appris. Je citerai Madame Hayate Khennouf et Messieurs Nicolas Marchand, Christophe Bérenguer, Nacim Meslem, Gildas Besançon, Didier George, Ahmad Hably et Christophe Prieur.

Un grand merci à mes collègues doctorants du labo, en particulier, mon colocataire de bureau **Humberto Stein Shiromoto** et ma chère **Oumayma Omar**. J'espère qu'on restera toujours amis et bonne chance à nous trois dans notre vie familiale et professionnelle.

Je dois encore des remerciements à toutes les personnes qui m'ont aidée, facilité et rendu mon séjour en France très agréable, je citerai, nos amis français, en particulier:

Ma marraine de cœur, **Annie Cluze**, « tu as réchauffé mon cœur quand il faisait froid », je n'oublierai jamais nos soirées et nos fous rires autour de bons desserts. Je tiens à t'assurer que **Gaby** est et sera toujours dans mon cœur, et que, j'aurais aimé qu'il soit là aujourd'hui. Je t'aime ma **Annie**.

La famille **Leclercq**, et tout particulièrement **Christiane** qui m'a toujours ouvert grands sa porte et son cœur. J'espère que tu trouveras dans ces mots un peu de l'affection que je vous porte.

La gérante de la résidence que j'ai occupée durant ces trois années **Madame Guderzo** ainsi que sa maman, pour leur accueil toujours chaleureux et pour leur gentillesse.

Je n'oublie pas d'exprimer ma réelle affection et toute ma gratitude au gardien de la résidence **Antoine**, qui a veillé sur moi comme sa fille et qui a rendu ma chambre très accueillante. Merci encore **Antoine** et sois sûr que je n'oublierai jamais ta bienveillance et ta sollicitude. Je te souhaite une longue vie et une excellente santé.

Sincèrement, **Sonia**

## **Résumé**

Notre travail de recherche concerne l'étude de la commande à base de modèles pour une sous-classe de systèmes dynamiques hybrides (SDH). L'outil de modélisation choisi est l'automate hybride rectangulaire (AHR) pour sa puissance d'analyse. Nous proposons ainsi une méthode pour la synthèse de la commande des SDH modélisés par des AHR. Cette méthode repose sur l'application d'une procédure amont/aval de commande hors-ligne qui détermine d'une façon maximale permissive les nouvelles gardes de transition de l'automate respectant des spécifications de commande imposées par l'utilisateur. Tous les calculs réalisés reposent sur la détermination de la durée de séjour, valeur contrainte par l'espace atteignable du sommet correspondant. La garde portant à la fois sur l'état continu et sur l'événement discret, la commande se fait par ce dernier car il s'agit du seul élément contrôlable. Nous nous intéressons alors à la construction du contrôleur temporisé autorisant l'occurrence des événements contrôlables du système dans un intervalle d'horloge défini au sens de la maximale permissivité.

## **Abstract**

In this thesis, we study the control of a class of hybrid dynamic systems (HDS). The chosen modeling tool is the rectangular hybrid automaton (RHA) for his analysis power. We propose a method for the control synthesis of HDS modeled with RHA. This method consists on the application of a downstream/upstream offline control procedure that determines in a maximal permissive way the new automaton transition guards respecting the desired control specifications. All computations are based on the determination of the duration of stay, a value constrained by the reachable space of the corresponding location. Since the guard refers to both continuous state and discrete event, the control is made by the latter because it is the controllable element. Then we are interested in the construction of the timed controller authorizing the system controllable event occurrence in a clock interval defined in a maximal permissive way.



# Table des matières

<b>Introduction générale</b> .....	1
<b>1 Systèmes hybrides</b> .....	6
1.1 Introduction.....	6
1.2 Dynamique des systèmes.....	7
1.2.1 Les Systèmes continus.....	7
1.2.2 Les Systèmes à Événements Discrets.....	8
1.2.3 Les Systèmes Dynamiques Hybrides .....	10
1.3 Modélisation des Systèmes Dynamiques Hybrides.....	11
1.3.1 Outils de modélisation des Systèmes à Evénements Discrets.....	12
1.3.2 Approches de modélisation des Systèmes Dynamiques Hybrides.....	15
1.3.3 Outils de modélisation de Systèmes Dynamiques Hybrides.....	17
1.4 Conclusion.....	25
<b>2 Commande par supervision</b> .....	27
2.1 Introduction.....	27
2.2 Commande par supervision : Théorie de base.....	28
2.3 Commande à base de modèles temporisés des Systèmes à Evénements Discrets	34
2.3.1 Commande par supervision en temps discret.....	34
2.3.2 Commande par supervision en temps continu.....	36
2.4 Commande des Systèmes Dynamiques Hybrides.....	44
2.4.1 Commande continue des Systèmes Dynamiques Hybrides.....	44
2.4.2 Commande événementielle des Systèmes Dynamiques Hybrides.....	45
2.5 Conclusion.....	49

---

<b>3 Synthèse de commande maximale permissive. Calcul pour un sommet.....</b>	<b>52</b>
3.1 Introduction.....	52
3.2 Présentation intuitive.....	53
3.2.1 Système tronçon de circulation.....	53
3.2.2 Modélisation.....	54
3.2.3 Spécifications de commande.....	55
3.2.4 Contrôlabilité des variables et commande temporisée.....	56
3.3 Les Automates Hybrides Rectangulaires.....	57
3.3.1 Syntaxe.....	57
3.3.2 Sémantique.....	60
3.3.3 Analyse d'atteignabilité d'un Automate Hybride Rectangulaire.....	61
3.4 L'automate contraint.....	65
3.4.1 Définitions.....	66
3.4.2 Problème d'espace convexe.....	67
3.4.3 Analyse d'atteignabilité de l'Automate Hybride Rectangulaire Contraint	68
3.5 Calcul local d'une commande maximale permissive.....	70
3.5.1 Cas d'une seule variable d'état.....	72
3.5.2 Application au système tronçon de circulation.....	75
3.5.3 Généralisation au cas de plusieurs variables d'état.....	76
3.6 Conclusion.....	79
<b>4 Synthèse de la commande dans le cas général.....</b>	<b>81</b>
4.1 Introduction.....	81
4.2 Procédure globale de commande .....	82
4.2.1 Généralisation de l'approche au cas de sommets « multi-sorties ».....	83
4.2.2 Algorithme de synthèse de commande maximale permissive.....	84
4.2.3 Application au système tronçon de circulation.....	86
4.3 Commande temporisée maximale permissive .....	88
4.4 Etude de cas.....	91
4.4.1 Système à deux tronçons de circulation.....	92
4.4.2 Système de production.....	98
4.5 Extension de la procédure de commande au cas d'événements incontrôlables....	103
4.6 Conclusion.....	110

---

<b>Conclusion générale et Perspectives.....</b>	<b>112</b>
<b>A Synthèse de commande du système à deux tronçons de circulation.....</b>	<b>116</b>
<b>B Étude d'une deuxième configuration du système de production.....</b>	<b>121</b>
<b>Bibliographie.....</b>	<b>125</b>



# Table des figures

1.1	<i>Chronogramme de l'évolution de l'état d'une vanne.....</i>	8
1.2	<i>Exemple de bac.....</i>	10
1.3	<i>Modèle à événements discrets correspondant au système bac.....</i>	10
1.4	<i>Exemple d'automate hybride.....</i>	19
1.5	<i>Exemple d'automate hybride linéaire.....</i>	21
1.6	<i>Exemple d'automate hybride rectangulaire.....</i>	22
1.7	<i>Exemple d'automate hybride rectangulaire initialisé.....</i>	22
1.8	<i>Modèle de RdP hybride d'un système de fabrication par lots.....</i>	24
2.1	<i>Le schéma de supervision.....</i>	29
2.2	<i>Le système de circulation.....</i>	32
2.3	<i>Tronçons à 2 états et procédé en isolation.....</i>	32
2.4	<i>Spécification.....</i>	32
2.5	<i>Modèle du contrôleur pour le système tronçon de circulation.....</i>	33
2.6	<i>Automate temporisé possible d'un tronçon de circulation.....</i>	36
2.7	<i>Spécification possible pour les tronçons de circulation.....</i>	37
2.8	<i>RdP à retards du système multimedia.....</i>	39
2.9	<i>Automate temporisé à retards du document multimédia.....</i>	40
2.10	<i>Superviseur multimédia.....</i>	40
2.11	<i>Portion d'un automate à temps continu.....</i>	41
2.12	<i>Approche d'Antsaklis.....</i>	46
2.13	<i>Approche de Dang.....</i>	48
2.14	<i>Approche de Kurovszky.....</i>	49

3.1	<i>Le système tronçon de circulation</i> .....	54
3.2	<i>Le modèle automate du tronçon de circulation</i> .....	54
3.3	<i>L'automate contraint du tronçon de circulation</i> .....	55
3.4	<i>Transitions continues et transitions discrètes d'un AHR</i> .....	61
3.5	<i>Analyse d'atteignabilité symbolique d'un AHR</i> .....	62
3.6	<i>Futur d'un polyèdre <math>\langle x \in [1, 3] \wedge y \in [0, 2] \rangle</math> sur une fonction de flux <math>\langle \dot{x} = 1 \wedge \dot{y} \in [1, 3] \rangle</math></i> .....	63
3.7	<i>Analyse en avant de l'automate tronçon de circulation</i> .....	65
3.8	<i>Spécification de commande</i> .....	66
3.9	<i>Portion d'un automate</i> .....	67
3.10	<i>Analyse en avant de l'automate contraint</i> .....	69
3.11	(a) <i>Espace <math>(q_{02})</math> de l'automate non contraint</i> – (b) <i>Espace <math>(q_{02})</math> de l'automate contraint</i> .....	70
3.12	<i>Calcul de la nouvelle garde <math>V_{i-1}</math></i> .....	71
3.13	<i>Principales variables nécessaires pour le sommet <math>(q_2)</math></i> .....	72
3.14	<i>Idée générale de commande maximale permissive</i> .....	76
4.1	<i>Calcul de la nouvelle garde <math>V_{i-1}</math></i> .....	82
4.2	<i>Partie d'un automate avec un sommet ayant plus d'une transition de sortie</i> .....	83
4.3	<i>Principales variables pour une transition <math>k</math> donnée à la sortie du sommet <math>(q_i)</math></i> .....	84
4.4	<i>Automate contraint atteignable du système tronçon de circulation</i> .....	87
4.5	<i>Superviseur optimal du système tronçon de circulation</i> .....	88
4.6	<i>Boucle générale de commande</i> .....	89
4.7	<i>Système à 2 tronçons de circulation</i> .....	93
4.8	<i>Automate contraint du système à 2 tronçons de circulation</i> .....	94
4.9	<i>Automate contraint atteignable du système à 2 tronçons de circulation</i> .....	95
4.10	<i>Forte interaction amont/aval</i> .....	96
4.11	<i>Superviseur temporisé maximal permissif du système à 2 tronçons de circulation</i> ...	97
4.12	<i>Système de production exclusive partagée</i> .....	98
4.13	<i>Le modèle automate du système de production</i> .....	99
4.14	<i>Automate contraint atteignable du système de production</i> .....	101
4.15	<i>Superviseur temporisé maximal permissif du système de production</i> .....	102
4.16	<i>Système général d'approvisionnement en eau</i> .....	105

4.17	<i>Système d’approvisionnement en eau.....</i>	106
4.18	<i>Le modèle automate système d’approvisionnement en eau.....</i>	107
4.19	<i>Automate contraint atteignable du système d’approvisionnement en eau.....</i>	109
A.1	<i>Calcul de la commande pour la transition <math>(q_{14}) \rightarrow (q_{04})</math>.....</i>	116
A.2	<i>Calcul de la commande pour la transition <math>(q_{04}) \rightarrow (q_{14})</math>.....</i>	117
A.3	<i>Calcul et vérification de la commande pour la transition <math>(q_{14}) \rightarrow (q_{04})</math> .....</i>	117
A.4	<i>Calcul de la commande pour la transition <math>(q_{13}) \rightarrow (q_{04})</math>.....</i>	118
A.5	<i>Nouvelles gardes pour le système à 2 tronçons de circulation.....</i>	119
B.1	<i>Nouvelle configuration du modèle automate du système de production.....</i>	122
B.2	<i>Nouvel automate contraint atteignable du système de production.....</i>	123
B.3	<i>Nouveau contrôleur temporisé maximal permissif du système de production.....</i>	124



# Introduction générale

La commande des systèmes automatisés représente un problème de grand intérêt pour la communauté scientifique. L'automatique traite différemment les problèmes de type continu et ceux de type séquentiels. Un ensemble de théories et de méthodes a été développé dans chacun de ces domaines afin de trouver des solutions performantes aux problèmes homogènes qui se posent, mais sans toujours intégrer les solutions et les apports de l'autre domaine.

Les procédés industriels sont complexes et afin de garantir leur bon fonctionnement, il est nécessaire de prendre en compte les aspects continus et événementiels de leur dynamique. D'une façon générale, les systèmes dynamiques faisant intervenir explicitement et simultanément des phénomènes ou des modèles de type dynamique continu et événementiel sont appelés *systèmes dynamiques hybrides* (SDH). Ces systèmes sont classiquement constitués de processus continus interagissant avec des processus discrets.

Les communautés des automaticiens et des informaticiens se sont intéressés à l'étude des systèmes hybrides, leur objectif étant d'apporter une solution en termes de modèle, de méthode, de performance et de qualité globale à des problèmes mal traités par les approches homogènes. Parmi ces problèmes, on peut citer les problèmes générés par des variations de structure en fonction des différents modes de marche d'un système, des discontinuités du fonctionnement des machines-outils ou en robotique, la modélisation des phénomènes transitoires rapides par une commutation des modèles et la description des procédés manufacturiers. Afin d'apporter une solution à ces problèmes, la recherche sur les SDH est concentrée autour de trois axes : la modélisation, l'analyse et la commande.

La *modélisation* cherche à formaliser des modèles précis pouvant décrire le comportement riche et complexe des SDH. Dans ce domaine, plusieurs formalismes ont été proposés afin d'établir un modèle homogène permettant la conciliation entre les parties discrètes et continues. Dans (Chen et Provan, 1997), les approches de modélisation des systèmes dynamiques hybrides sont classées en trois catégories : (i) l'approche discrète qui consiste à approximer les dynamiques continues de façon à se ramener à un système à événements discrets : (ii) l'approche continue qui revient à approximer les dynamiques discrètes par des systèmes continus afin d'utiliser la théorie de ces derniers et (iii) l'approche mixte qui considère à la fois les comportements continus et discrets dans une même structure. Cette dernière approche possède l'intérêt de ne faire aucune abstraction d'informations sur le système à étudier. L'approche de modélisation à laquelle nous nous sommes intéressés dans le cadre de notre travail fait partie de cette dernière catégorie de modèles et considère le modèle du système hybride comme une extension de l'automate discret en associant une dynamique continue à chaque état discret. La composante continue est décrite par un ensemble d'équations différentielles et la composante discrète est décrite par un automate fini. Le modèle résultant de cette approche est connu comme étant le modèle *automate hybride* (Henzinger, 1996).

L'*analyse* porte sur le développement des outils de simulation, de validation et de vérification des SDH. Les problèmes rencontrés sont liés à la complexité de cette analyse et à l'interprétation physique de certaines propriétés à analyser telles que la stabilité globale du système à travers ses phases consécutives de fonctionnement. Un problème central dans la vérification des propriétés des systèmes hybrides modélisés par des automates hybrides est l'analyse d'atteignabilité. Dans ce travail, nous nous intéressons uniquement à l'analyse d'atteignabilité en avant que l'on réalise grâce à l'utilisation du logiciel PHAver (Frehse, 2005 ; Asarin et al., 2006). Celle-ci donne tous les espaces atteignables de l'automate étudié et permet ainsi de modéliser l'*automate atteignable*.

La *commande* concerne la synthèse de contrôleurs discrets ou hybrides conformément à certains objectifs de performance et de sûreté de fonctionnement du procédé hybride commandé. Certaines approches de la commande des SDH porte sur la formulation et la résolution d'un problème de commande optimale caractéristique des systèmes continus. D'autres approches visent à rechercher une stratégie discrète permettant de restreindre le comportement du SDH pour satisfaire des spécifications imposées.

Notre travail de recherche concernant la synthèse de la commande s'inscrit dans cette dernière catégorie. L'objectif est d'apporter une solution originale au problème de la synthèse optimale de la commande pour les systèmes dynamiques hybrides. A cet effet, nous proposons une démarche basée sur une extension de la théorie classique de la commande supervisée, développée par Ramadge et Wonham (Ramadge et Wonham, 1987*b*, 1989) pour les systèmes à événements discrets.

Notre but est de déterminer une commande d'une classe de systèmes dynamiques hybrides, à forte composante événementielle, modélisés par un automate hybride rectangulaire. On étudie en particulier les systèmes à flux continu dans lesquelles les variables d'état sont non contrôlables et non observables, en opposition à l'événement discret qui est à la fois contrôlable et observable. Il s'en suit alors que l'on s'intéresse à un pilotage discret du système hybride.

Par conséquent, tout l'intérêt de notre travail réside dans le fait d'imposer des contraintes sur l'état non contrôlable en agissant sur l'événement discret, qui lui est contrôlable. Ces contraintes sur les variables d'état du système, appelées aussi spécifications de commande, sont ajoutées au niveau des gardes de transitions de l'automate. Notre objectif est de trouver le comportement permettant le respect de ces spécifications. Il faudra autant que possible sélectionner la meilleure solution au sens de la maximale permissivité.

Ce mémoire est organisé en quatre chapitres.

Le premier chapitre permet d'exposer le contexte de notre étude, à savoir les systèmes dynamiques hybrides (SDH), objet de notre travail. Ces systèmes provenant d'une combinaison entre les systèmes continus et les systèmes à événements discrets (SED), une introduction brève de ces deux derniers systèmes sera donnée en amont de la présentation des systèmes hybrides. Les différents outils de modélisation de ces systèmes hybrides seront développés par la suite pour pouvoir introduire les automates hybrides rectangulaires, modèle choisi dans le cadre de notre approche de contrôle.

Dans le deuxième chapitre, nous présentons un état de l'art sur les principales approches de commande des systèmes hybrides. Pour ce faire, nous exposons d'abord le concept de la commande par supervision. Puis nous décrivons les différentes approches de commande des SED, des SED temporisés, pour enfin introduire celles des SDH qui nous intéressent dans nos travaux de recherche. L'accent est mis sur la synthèse de commande basée sur l'outil

automate (automate à états finis, automate temporisé et automate hybride) car il s'agit du modèle utilisé dans notre propre approche de synthèse.

Le troisième chapitre présente notre démarche de commande des systèmes dynamiques hybrides (SDH) calculée pour un sommet donné de l'automate. D'abord nous rappelons le cadre de modélisation considéré dans notre contribution : les automates hybrides rectangulaires ainsi que les méthodes permettant l'analyse comportementale de ce modèle. Nous détaillons par la suite notre démarche de commande de ces systèmes. Celle-ci commence par la caractérisation des spécifications de commande que l'on souhaite imposer au système. Elles sont traduites par des contraintes ajoutées aux gardes de transition de l'automate. Notre approche de commande se fait alors par une procédure amont/avale permettant le calcul de nouvelles gardes respectant ces contraintes.

Le quatrième chapitre est consacré à la généralisation de notre approche à l'automate global. On y expose dans un premier temps un algorithme de calcul de toutes les nouvelles gardes de transition de l'automate pouvant avoir la structure la plus complexe possible. Par la suite, on détaille la procédure de calcul hors-ligne du superviseur temporisé maximal permissif qui autorise l'occurrence des événements contrôlables du système par des valeurs d'horloge déterminées par programmation linéaire. On obtient toutes les trajectoires possibles du système contraint. Enfin, une extension possible de notre approche est proposée. Il s'agit de la prise en compte des événements incontrôlables. Notre procédure générale ainsi que son extension sont illustrées par plusieurs exemples à travers le chapitre.



# Chapitre 1

## Systemes Hybrides

### 1.1 Introduction

L'étude des systèmes dynamiques a porté, pour la plupart, sur les deux domaines distincts des *systèmes continus* et des *systèmes à événements discrets* (SED). Les premiers sont généralement modélisés et analysés en utilisant les équations différentielles pour saisir le comportement continu des différentes variables, alors que les SED sont pilotés par des événements pour saisir le comportement en états discrets. Récemment est apparu un intérêt pour le comportement dynamique des systèmes qui allie ces deux classes. Ces systèmes sont appelés *systèmes hybrides*. La motivation pour l'examen des systèmes ayant des dynamiques mixtes existe aux niveaux pratique et théorique. Ces dernières années, il y a eu une hausse rapide de l'intérêt porté à l'étude des systèmes hybrides. La plupart des travaux effectués ont abordé la question de la modélisation du comportement hybride et de l'analyse des performances. D'autres efforts ont été réalisés pour aborder la commande de ce type de systèmes.

L'objectif de ce chapitre est de présenter le contexte de notre étude, à savoir les systèmes hybrides, objet de notre travail. Ces systèmes provenant d'une combinaison entre les systèmes continus et les systèmes à événements discrets, une introduction brève de ces deux derniers systèmes sera donnée en amont de la présentation des systèmes hybrides. Les différents outils de modélisation de ces systèmes hybrides seront développés par la suite pour pouvoir

introduire les automates hybrides rectangulaires, modèle choisi dans le cadre de notre approche de contrôle.

### 1.2 Dynamique des systèmes

Afin de répondre aux besoins croissants en termes de productivité d'un système industriel, de fluidité de la circulation dans un réseau urbain, de disponibilité des systèmes de communication, etc., il a été nécessaire de mettre en place des techniques puissantes de modélisation, d'analyse et de commande de ces systèmes. L'objectif est d'augmenter les cadences de production, le flux de circulation et la vitesse de communication.

Il existe plusieurs types d'abstractions des systèmes, selon les objectifs qu'on se donne et la réalité physique du système en question. On peut être amené à choisir tel ou tel type de modèle par rapport à la qualité d'approximation ou à la facilité de calcul qu'il permet. Par exemple, la modélisation d'un système de trafic urbain peut être perçue à travers plusieurs points de vue:

- Soit par un système discret qui est précis mais peut mener à des calculs complexes
- Soit par une approximation continue qui peut être satisfaisante en permettant des calculs plus simples comme nous le verrons dans la suite de cette thèse.

La caractérisation des systèmes peut se faire selon la dynamique qui exprime leur fonctionnement. Trois abstractions sont possibles pour modéliser cette dynamique: les systèmes continus, les systèmes à événements discrets et les systèmes dynamiques hybrides (Zaytoon, 2001).

#### 1.2.1 Les Systèmes continus

Les systèmes continus sont constitués d'éléments caractérisés par une ou plusieurs variables qui peuvent prendre des valeurs réelles lorsque le temps, lui-même « continu », évolue. On peut citer comme exemple de variables, une position, une vitesse, une accélération, un niveau, une pression, une température, un débit, une tension, etc.

Afin de gérer ce type de système, on fait appel à des outils mathématiques aptes à la représentation de la dynamique continue, par exemple les équations différentielles assorties de diverses transformations (Laplace, Fourier...), les méthodes d'état à forme matricielle.

L'automaticien perçoit les systèmes continus à travers une représentation reposant le plus souvent sur des variables d'état continues et une variable temporelle.

### 1.2.2 Les Systèmes à Evénements Discrets

Lorsque les espaces d'état d'un système sont décrits naturellement par un ensemble discret tel que  $\{0, 1, 2, \dots\}$  et que les transitions d'un état à un autre sont observées uniquement à des instants discrets dans le temps, on associe à ces transitions d'état des événements et on appelle ce type de système, Système à Evénements Discrets (SED) (Cassandras et Lafortune, 2008).

Au lieu de s'intéresser au déroulement continu des phénomènes, les modèles SED ne se soucient que des débuts et des fins de ces phénomènes (les événements discrets) et de leur enchaînement logique, dynamique ou temporel. Ces systèmes recouvrent un grand nombre de situations, allant de la circulation de véhicules en réseau urbain au fonctionnement de machines dans un atelier flexible.

Par exemple, considérons une vanne qui peut prendre 4 états : « fermée », « en ouverture », « ouverte » et « en fermeture ». Une évolution possible de ce SED est représentée dans le chronogramme de la figure 1.1.

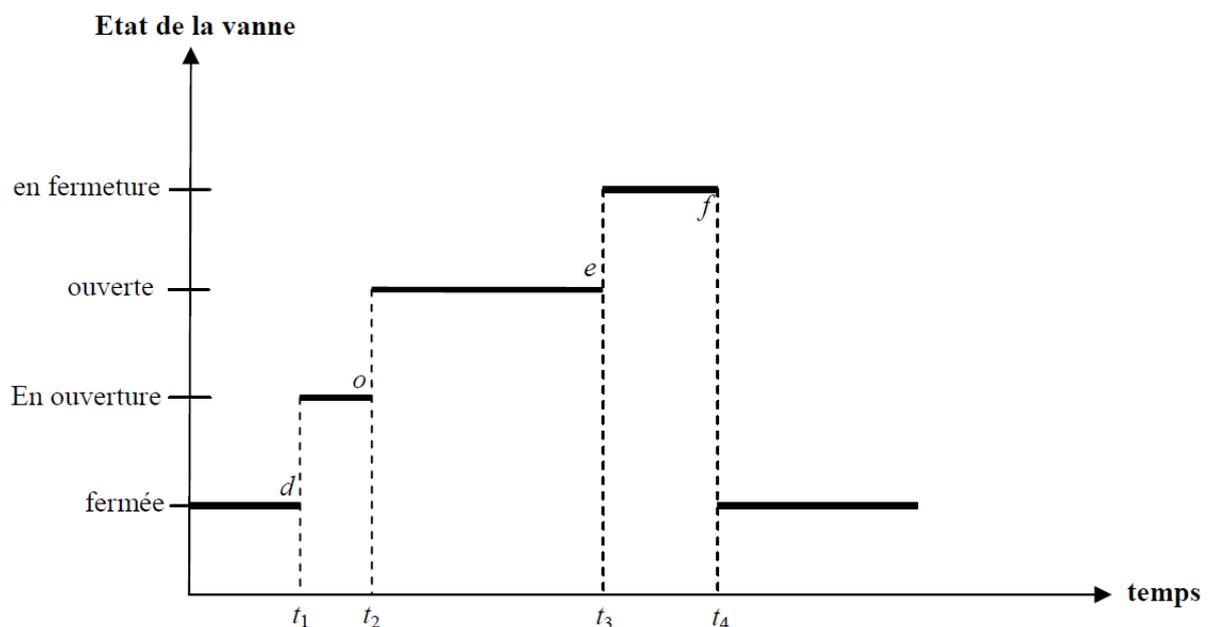


Fig.1.1- Chronogramme de l'évolution de l'état d'une vanne.

Dans son état initial, la vanne est supposée fermée. A l'instant  $t_1$ , l'événement  $d =$  'début d'ouverture de la vanne' se produit. De façon similaire, le système évolue aux instants  $t_2$ ,  $t_3$  et

$t_4$  sur occurrence des événements ‘fin d’ouverture’, ‘début de fermeture’ et ‘fin de fermeture’ représentés respectivement par les symboles  $o$ ,  $e$  et  $f$ .

Les instants d’occurrence des événements peuvent se produire à n’importe quel moment (il n’y a pas d’échantillonnage).

Par abus de langage, on peut parler indifféremment du symbole comme de l’événement physique qu’il représente. Ainsi, on dira que l’événement  $d$  modélise la transition de l’état « fermée » à l’état « en ouverture ».

L’évolution d’un SED peut être décrite de deux manières :

- Soit par un *modèle temporisé*, c’est-à-dire un ensemble de couple :  $(e, t)$  où ‘ $e$ ’ représente un événement et ‘ $t$ ’ représente l’instant d’occurrence de cet événement. Dans notre exemple, l’évolution de l’état de la vanne peut être définie par les couples:  $(d, t_1)$ ,  $(o, t_2)$ ,  $(e, t_3)$ ,  $(f, t_4)$ ,... Cet ensemble ordonné de couples constitue ce qu’on appelle une *séquence*. Une telle description se place à un niveau temporel dans le sens où l’instant d’occurrence des événements est une information considérée comme pertinente.
- Soit par un *modèle logique* où seul l’ordre d’occurrence des événements importe. Dans ce cas, le temps est omis et on parle de séquences d’événements. Dans ce contexte, le fonctionnement de la vanne est décrit par la séquence des événements :  $d, o, e, f, \dots$ . Les *modèles logiques* sont utilisés pour l’étude des propriétés qualitatives des SED.

Généralement, les SED peuvent avoir un comportement non déterministe, dans le sens où l’on ne peut prévoir à priori quelle sera l’évolution du système. En d’autres termes, pour un état donné du système, plusieurs événements différents sont supposés susceptibles de se produire. Par conséquent, une séquence unique ne suffit plus pour décrire le comportement du système, contrairement à l’exemple de la vanne. L’évolution d’un SED sera alors décrite, en général, par un ensemble de séquences d’événements, qui constitue un *langage* sur l’ensemble des événements possibles dans le système (Hopcroft et al., 2007).

On peut citer parmi les modèles généraux de modélisation des SED :

- Les automates temporisés et les Réseaux de Petri temporels qui sont des modèles temporisés.
- Les automates à états finis et les réseaux de Petri qui sont des modèles logiques.

### 1.2.3 Les Systèmes Dynamiques Hybrides

Comme nous venons de le voir, les systèmes automatisés sont traditionnellement représentés soit par un modèle à dynamique continue, soit par un modèle à événements discrets. Un même système peut être décrit par l'un ou l'autre des modèles selon la vue, le temps d'observation et/ou les phénomènes auxquels on s'intéresse. Prenons l'exemple d'un bac qui peut être rempli ou vidé (figure 1.2.).

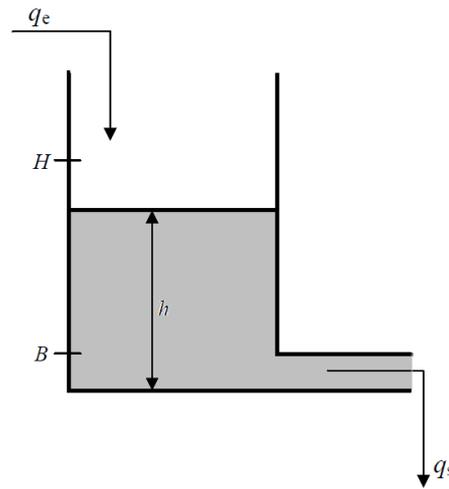


Fig.1.2- Exemple de bac.

- Si on s'intéresse au phénomène de remplissage du bac afin de modéliser le niveau d'eau qu'il contient, le modèle correspondant est un modèle continu où le système physique peut être décrit par l'équation :  $S \frac{dh(t)}{dt} = q_e(t) - u(t) \cdot \alpha h(t)$ ; où  $q_e(t)$  est le débit d'entrée,  $h(t)$  la hauteur d'eau,  $u(t)$  l'état de la vanne qui prend la valeur 0 pour fermée et 1 pour ouverte,  $S$  la section du bac et  $\alpha$  un paramètre.
- Si une description des états remplissage ou évacuation est suffisante, on s'orientera vers un modèle SED, qui peut être décrit par l'automate de la figure 1.3.

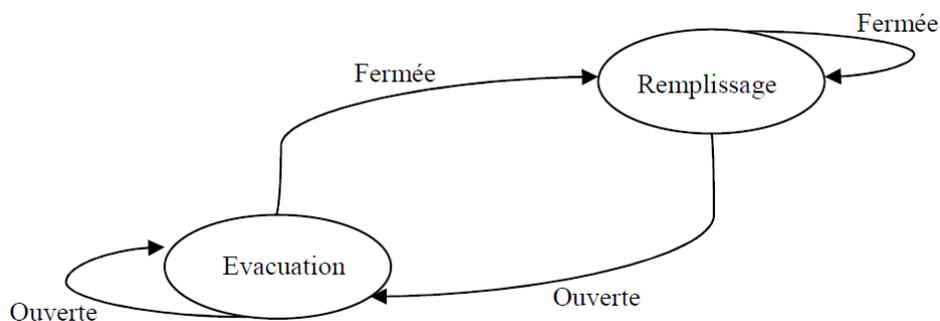


Fig.1.3- Modèle à événements discrets correspondant au système bac.

Dans certains cas, l'approximation qui consiste à classer un système dans l'une des deux catégories citées peut être trop forte. Il est alors nécessaire d'utiliser un modèle dynamique permettant la prise en compte des variables d'état continues et discrètes. Ce type de système est appelé Système Dynamique Hybride (SDH). Il existe plusieurs domaines d'applications pour ce type de système; on peut citer les procédés *batch*, les systèmes électrotechniques, les réseaux de communication, les systèmes de transport, la commande des procédés industriels, les systèmes flexibles de production, l'industrie agro-alimentaire, la commande des moteurs biologiques, la robotique, etc.

Les SDH évoluent dans le temps et combinent des variables continues et des variables discrètes (Zaytoon, 2001). Un état discret du système peut être vu comme un système continu avec des variables continues reliées par des contraintes. Cependant, la portée de ces contraintes est restreinte à l'état en question. La transition du système d'un état à un autre fait changer son mode de fonctionnement en lui faisant subir d'autres lois continues propres au nouvel état.

Ce modèle est celui qui est retenu dans nos travaux de recherche. On le présentera plus en détails dans le chapitre suivant.

### 1.3 Modélisation des Systèmes Dynamiques Hybrides

Comme nous l'avons mentionné dans le paragraphe précédent, il existe de nombreux systèmes réels qui évoluent selon des sous processus continus qui sont démarrés et arrêtés par des commandes discrètes en relation avec les sous-processus continus. Dans ce cas, les procédés ont rarement un comportement purement discret ou purement continu mais plutôt une combinaison des deux. Ces *Systèmes Dynamiques Hybrides* (SDH) peuvent être de natures très diverses. On trouve des systèmes continus auxquels sont associés des commutations discrètes et des systèmes à événements discrets auxquels sont associées certaines évolutions continues.

Dans nos travaux de recherche, on se place dans le cas où l'on ajoute des dynamiques simples du type horloge ou du type ( $\dot{x} = constante$ ),  $\dot{x}$  étant la dérivée de la variable  $x$  par rapport au temps. C'est pour cela que tous nos résultats et approche seront inspirés des techniques provenant des domaines des SED.

Dans la littérature, plusieurs outils de modélisation des systèmes hybrides ont été proposés. Parmi ces outils, on cite les automates hybrides (Alur et al., 1993), les automates hybrides rectangulaires (Henzinger et al., 1998), les automates hybrides linéaires (Müller et Stauner,

2000), les RdP hybrides (Alla et David, 1998 ; David et Alla, 2010), les statecharts hybrides (Harel, 1987 ; Harel et al, 1987 ; Mendler et Lüttgen, 2001), les bond graphs hybrides (Mosterman, 1997), etc.

Dans ce qui suit, nous allons insister sur les outils dans lesquels nous avons trouvé des réponses aux objectifs que nous nous sommes fixés. Les outils cités ci-dessus provenant en général d'une extension des outils de modélisation des SED, nous avons jugé important de présenter, du moins brièvement, ces derniers, avant d'aborder la modélisation des SDH.

### 1.3.1 Outils de modélisation des Systèmes à Événements Discrets

Dans ce qui suit, nous présentons quelques outils de modélisation des SED. Comme nous l'avons mentionné dans la section précédente, Ces outils peuvent être classés en deux catégories : les modèles logiques et les modèles temporisés.

#### 1.3.1.1 Les modèles logiques

Ces modèles prennent en compte l'ordre logique d'occurrence des événements dans un SED. Par conséquent, le temps est décrit implicitement dans une trajectoire, et seul l'ordre d'occurrence des événements constituant une trajectoire est pris en considération. Ces modèles sont exploités dans le but d'étudier les propriétés qualitatives des SED. On cite parmi ces modèles les automates à états finis et les réseaux de Petri.

##### Les automates à états finis

Un *automate à états finis* constitue une machine à états qui permet de décrire les évolutions possibles d'un système à événements discrets (Cassandras et Lafortune, 2008). Le comportement d'un SED est alors représenté à travers un ensemble d'événements associés à un ensemble d'états. D'une façon formelle, un automate à états finis est défini par un quintuplet :

$$G=(Q, \Sigma, \delta, q_0, Q_m)$$

Où :

- $Q$  est un ensemble fini d'états ;
- $\Sigma$  est un ensemble fini d'événements (ou de symboles) ;

- $\delta$  est une fonction de transition,  $\delta: Q \times \Sigma \rightarrow Q$  ;
- $q_0 \in Q$  est un état initial ;
- $Q_m \subseteq Q$  est l'ensemble d'états finaux (états marqués).

On dit d'un automate à états finis qu'il est *déterministe*, si à partir d'un état donné, au plus une seule transition est possible sur l'occurrence d'un événement.

### Les Réseaux de Petri

Le modèle Réseau de Petri (RdP), introduit en 1964 par C. A. Petri (Petri, 1962), représente un outil de modélisation de SED particulièrement adapté pour spécifier le comportement des systèmes SED de grande taille. Ce modèle permet de modéliser et de visualiser des primitives de comportement telles que la synchronisation, le parallélisme ou le partage de ressources (David et Alla, 1989). Les RdPs possèdent à la fois une description graphique et mathématique. Ils sont représentés par des places et des transitions, reliées par des arcs.

Il existe plusieurs extensions des RdPs. On peut citer les RdPs colorés (David et Alla, 1989), et les RdPs temporels (Merlin, 1974; Berthomieu et Diaz, 1991). Grâce à cette multitude d'extensions, les RdPs peuvent s'appliquer à la plupart des phases de développement d'un système, de la spécification de la commande à la supervision en passant par le diagnostic et la validation (Philippot, 2006).

#### 1.3.1.2 Les modèles temporisés

L'information temporelle est, dans certaines applications, indispensable et doit être considérée explicitement dans le modèle du SED. Les modèles ayant cette propriété sont dits temporisés. On peut citer, parmi ces modèles, les automates temporisés et les RdPs temporels.

#### Les automates temporisés

Ce sont des automates classiques munis d'un ensemble de variables réelles, appelées *horloges*, qui évoluent de manière continue et synchrone avec le temps (Alur et Dill, 1994). A chaque transition, on associe :

- une condition sur la valeur des horloges, dite *garde*, qui décrit le moment où la transition peut être exécutée
- un ensemble d'horloges remises à zéro lors du franchissement de la transition.

Chaque état discret possède un invariant (une contrainte sur les horloges) qui restreint le temps d'attente dans l'état et par conséquent, force l'exécution d'une transition.

Formellement, un *automate temporisé* est défini par un septuplet :

$$A=(Q, X, \Sigma, E, q_0, Q^f, I)$$

Où :

- $Q$  est un ensemble fini de sommets;
- $X= \{x_1, \dots, x_n\}$  est un ensemble fini d'horloges ;
- $\Sigma$  est un ensemble fini d'événements (ou de symboles ou d'actions) ;
- $E$  est une relation de transition entre les sommets ;
- $q_0 \in Q$  est le sommet initial ;
- $Q^f \subseteq Q$  est un sous- ensemble de sommets finaux qui n'admettent pas de transitions de sortie vers d'autres sommets (sommets puits) ;
- $I$  est une fonction qui associe à chaque sommet une contrainte, appelée *invariant*.

Le modèle automate temporisé constitue un formalisme relativement simple à manier. Cependant, il a l'expressivité nécessaire pour la modélisation des systèmes temporisés. Ce formalisme offre aussi une capacité importante d'analyse comportementale des systèmes à travers les différentes méthodes développées dans la littérature, comme le model-checking (Yovine, 1998) ou l'analyse d'atteignabilité (Alur, 1999 ; Bengtsson et Yi., 2004).

### Les réseaux de Petri temporels

Il s'agit d'une extension du modèle réseau de Petri qui associe deux dates *min* et *max* à chaque transition (Merlin, 1974). Si une transition  $t$  a été sensibilisée pour la dernière fois à une date  $\theta$ , alors  $t$  ne peut pas être franchie avant la date  $\theta + min$  ou après la date  $\theta + max$ , sauf si une autre transition a désensibilisé  $t$  avant que celle-ci ne soit franchie. Ces modèles expriment des spécifications « en délais » qui sont largement utilisés dans la description de l'évolution temporelle des processus industriels. Dans le but d'utiliser la grande capacité

d'analyse représentée par le modèle automate temporisé, plusieurs travaux se sont intéressés à la transformation du modèle RdP temporel en modèle automate temporisé (Sava et Alla, 2001 ; Cassez et Roux, 2006).

### 1.3.2 Approches de modélisation des Systèmes Dynamiques Hybrides

Alors que dans le paragraphe précédent nous nous sommes intéressés à la modélisation des SED logiques et temporisés, nous présentons ci-dessous les Systèmes Hybrides ou la dynamique continue va au-delà de l'horloge.

L'existence de plusieurs outils de modélisation des SDH exprime la difficulté du choix adéquat de la méthode de modélisation. Le principal critère de sélection est lié à la problématique considérée.

Généralement, un SDH est modélisé par un ensemble de systèmes à dynamique continue interagissant avec un ou plusieurs systèmes à événements discrets (Kurovsky, 2002). Le point commun qui existe entre ces formalismes est que les événements discrets interfèrent dans l'évolution continue et que les modèles nécessitent à la fois des variables d'état continues et discrètes.

Les approches de modélisation des SDH peuvent être classées en trois catégories principales (Chen et Provan, 1997) : l'approche continue, l'approche événementielle et l'approche mixte.

#### **Approche de modélisation continue**

Cette approche consiste à étudier le comportement des modèles continus en présence des discontinuités, et éventuellement de définir un modèle « étendu ». Il s'agit de définir une approximation des dynamiques discrètes du système hybride par des équations différentielles pour modéliser l'occurrence des événements discrets. L'idée est, qu'en utilisant une approche unifiée dans le domaine des systèmes continus, où les théories sont bien établies, les questions de stabilité, de commandabilité et d'observabilité pourront être étudiées selon les théories classiques.

Plusieurs travaux dans la littérature traitent de la modélisation des systèmes hybrides par une approche continue. Ces travaux peuvent être classés en deux catégories :

- Une partie orientée vers la définition d'ensemble de transitions afin de pouvoir considérer l'évolution discrète du système hybride (Witsenhausen, 1966 ; Tavernini, 1987 ; Branicky et al., 1994).
- Une partie basée sur l'introduction des variables supplémentaires dans le modèle continu, représenté par des équations différentielles, pour décrire le fonctionnement des systèmes hybrides (Brockett, 1993, 1994).

L'avantage de l'approche continue est que l'on revient à des méthodes classiques d'analyses des systèmes continus linéaires ou non linéaires. Ses deux principaux inconvénients sont la complexité des équations obtenues ainsi que le fait de ne pas représenter explicitement l'évolution discrète pour l'utilisateur.

### **Approche de modélisation événementielle**

Avoir une approche purement discrète pour modéliser les systèmes hybrides consiste à supprimer les dynamiques continues ou à faire une approximation de l'évolution continue de façon à ce que le système hybride soit représenté uniquement par les événements qui le caractérisent. La modélisation événementielle d'un SDH pourra alors permettre de faire appel à la théorie classique des SED.

Les travaux de Puri (Puri et al., 1996) présentent une méthode directe afin d'obtenir un modèle événementiel temporisé du système hybride, qui consiste à découper l'espace d'état continu en plusieurs régions, associées chacune à un état discret. Cependant, ce concept de modélisation reste confronté au compromis entre la précision et le nombre d'états discrets rapidement croissant de manière exponentielle.

### **Approche de modélisation mixte**

Les approches d'intégration des aspects hybrides dans des modèles continus ou événementiels ne permettent pas une représentation explicite de l'interaction entre la partie continue et la partie événementielle.

Dans une approche mixte, chacune des deux composantes (continue et discrète) est représentée de manière explicite et rigoureuse et leur collaboration est prise en compte dans l'interface qui les relie.

L'approche mixte repose sur l'hypothèse que le fonctionnement d'un système hybride est une séquence de deux phases :

- La première phase est une évolution du modèle continu des variables au cours du temps pouvant valider certaines transitions.
- La seconde phase est une évolution du modèle discret qui engendre alors la mise en place d'un nouvel état discret qui se traduit par l'élaboration d'un nouveau système continu.

Par conséquent, les modèles développés dans le cadre de cette approche reposent sur l'interaction de deux sous-modèles :

- Le premier concerne les aspects événementiels. Il est basé sur les *automates à états finis*, les *réseaux de Petri* ou des extensions de ces formalismes.
- Le deuxième est formalisé par des équations d'état pour les aspects continus.

Ainsi, chacun des aspects, continu ou événementiel, est décrit sous une forme classique et l'aspect hybride est clairement considéré. L'aspect événementiel influe sur le modèle continu par la validation de certaines des équations continues en fonction de l'état discret actif et l'aspect continu agit sur le modèle événementiel en validant ou en forçant le franchissement de certaines transitions.

L'approche mixte a engendré l'apparition de nouveaux modèles hybrides à partir des modèles continus et discrets. On peut citer comme exemples, les *réseaux de Petri hybrides* et les *automates hybrides*, qui sont obtenus à partir des modèles discrets, et les MLD (*Mixed Logical Dynamical*) obtenus par des modèles continus (Bemporad et Moriari, 1999). Cette dernière méthode est basée sur l'idée d'introduire des variables auxiliaires qui permettent de modéliser les relations existantes entre les parties continue et discrète. Par conséquent, le passage à la partie discrète nécessite l'ajout de variables logiques. Concernant la partie correspondant au passage du discret au continu, des variables auxiliaires sont ajoutées.

### 1.3.3 Outils de modélisation des Systèmes Dynamiques Hybrides

Dans la suite, nous allons présenter quelques modèles de l'approche mixte rencontrés dans la littérature.

### 1.3.3.1 Les automates hybrides

Les *automates hybrides* (Alur et al., 1993), représentent une extension des automates à états finis. Ce sont des systèmes qui intègrent deux composantes :

- Une composante ayant un comportement discret, naturellement modélisée par un automate à états finis
- Une composante dont le comportement varie de façon continue dans le temps, modélisée par un système d'équations différentielles.

L'évolution d'un *automate hybride* est caractérisée par une alternance de pas continus, où les variables d'état et le temps progressent de façon continue, et de pas discrets où plusieurs transitions discrètes et instantanées peuvent être franchies.

Il est intéressant de noter qu'un *automate temporel* (Alur et Dill, 1994) correspond à un automate hybride particulier où toutes les variables continues admettent des dérivées par rapport au temps égales à 1.

Ainsi, d'un point de vue informel et général, un automate hybride apparaît ainsi comme un automate à états finis pilotant un ensemble d'équations différentielles qui modélise la dynamique continue du système.

Le modèle automate hybride se compose d'un ensemble fini de variables, noté  $X$ , et d'un graphe orienté, noté  $(S, E)$ , où l'ensemble  $S$  est composé par les sommets du graphe et les éléments de l'ensemble  $E$  représentent les transitions discrètes. L'état de l'automate change instantanément à l'occurrence d'un événement discret ou par la validation d'une condition logique spécifiée sur la valeur de la variable continue (Henzinger, 1996).

Soit l'automate illustré dans la figure 1.4 modélisant un système hybride. Dans ce modèle:

- l'évolution continue est représentée par des équations différentielles associées aux sommets du graphe.
- l'évolution événementielle est modélisée par les arcs étiquetés des graphes.

Les sommets « Eteint » et « Allumé » représentent les états discrets du système où l'évolution continue a lieu.  $x$  représente une horloge alors que  $y$  correspond à une variable plus complexe.

Les prédicats  $x \in [2, 4]$  et  $x \in [5, 10]$  sur les arcs reflètent les conditions, dites aussi *gardes*, pour l'occurrence des événements *allumer* et *éteindre*, respectivement.

Les prédicats  $x \leq 4$  et  $x \leq 10$  dans les sommets constituent les *invariants* de l'automate, c'est-à-dire, des conditions imposées aux variables continues du système pour rester dans un état discret (dans notre cas, les états « Eteint » ou « Allumé »).

L'arc d'entrée au sommet d'origine « Eteint » symbolise l'état initial du système. L'étiquette de cet arc  $x := [5, 10] \wedge y := 0$  décrit la région de l'espace continue à partir de laquelle la dynamique du système hybride démarre.

L'évolution des variables  $x$  et  $y$  dans le sommet « Eteint », respectivement le sommet « Allumé », se fait conformément aux équations différentielle  $\dot{x} = 1$  et  $\dot{y} = -4y$ , respectivement  $\dot{x} = 1$  et  $\dot{y} = 3y + 2$ , appelées *conditions de flux* ( $\dot{x}$  désigne la dérivée du premier ordre par rapport au temps de la variable  $x$ ).

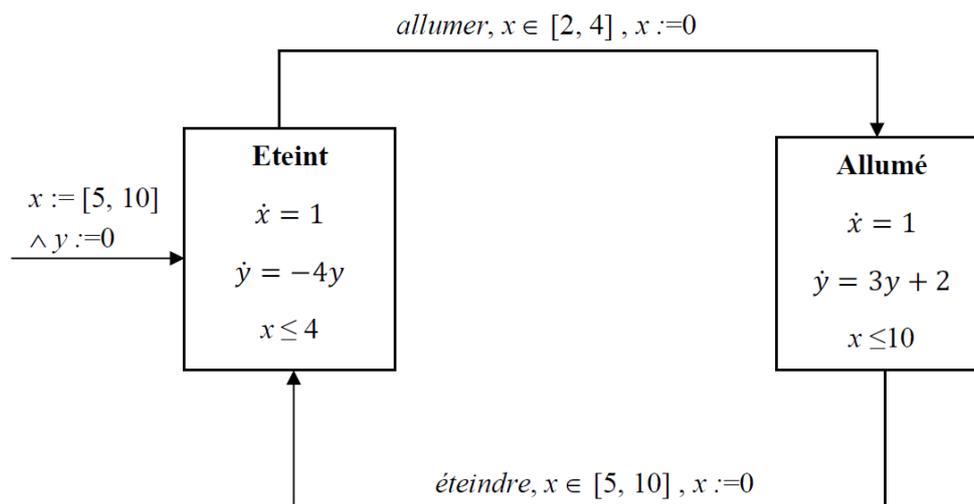


Fig.1.4- Exemple d'automate hybride.

L'analyse et la commande des systèmes à travers l'utilisation des automates hybrides occupent une place considérable dans l'étude des SDH. Il est question de vérifier certaines propriétés qualitatives et quantitatives sur un système grâce à l'exploitation des méthodes relevant de l'informatique et de l'automatique. La commande se base, d'abord, sur une modélisation de SDH à vérifier par un automate hybride. Elle repose ensuite, sur l'application d'une analyse des propriétés à travers les techniques de model-checking (Henzinger et al., 1997), et à travers l'analyse d'atteignabilité (Alur et al. , 1993, 1995).

L'analyse d'atteignabilité constitue un problème central dans la vérification des propriétés des systèmes hybrides modélisés par des automates hybrides. Elle nous permet de mieux

connaître l'évolution du système dans sa globalité. Ainsi, les informations concernant le comportement du système hybride, obtenues à cette étape, seront ensuite utilisées lors de la synthèse de la commande comme nous le verrons dans les chapitres suivants. L'objectif est de retenir toutes les informations concernant l'évolution du système.

L'analyse d'atteignabilité consiste à trouver l'espace d'état atteignable par l'évolution du système hybride étudié. Cependant, ce problème n'est pas décidable pour un automate hybride sans hypothèses particulières (Henzinger et al. 1998). Cette terminologie vient de l'informatique où il y a mise en œuvre des propriétés. Un problème est non décidable s'il n'existe pas d'algorithme permettant la vérification des propriétés en un temps fini.

Par conséquent, de nombreux travaux ont défini des restrictions afin d'obtenir des sous-classes pour lesquelles certaines propriétés sont décidables (Derbel, 2009).

### 1.3.3.2 Les sous-classes d'automates hybrides

Dans la littérature, plusieurs sous-classes du modèle automate hybride ont été étudiées afin de simplifier les dynamiques du modèle initial. Ceci a pour but de faciliter l'analyse et la commande du modèle en question tout en maintenant un domaine d'applicabilité des approches proposées. Parmi les modèles proposés, on peut citer : les automates hybrides linéaires ou les automates hybrides rectangulaires. Les sous-classes du modèle automate hybride sont décrites ici d'une manière informelle. Le modèle automate hybride rectangulaire, constituant la base de notre travail de recherche, sera présenté et décrit d'une façon plus détaillée et formelle dans les chapitres qui suivent.

#### Les automates hybrides linéaires

Un *automate hybride linéaire* (Alur et al. 1993, 1997, 2003 ; Henzinger et Rusu, 1998) est un graphe de contrôle avec des conditions sur les variables réelles. Chaque sommet du graphe représente un mode d'exploitation du système, et il est défini par des équations différentielles qui décrivent les évolutions (flux) possibles des variables réelles lorsque le système reste dans le même mode. Chaque sommet est aussi défini par une condition d'invariant: pour que le système puisse séjourner dans le même sommet de l'automate hybride linéaire, les variables doivent satisfaire les invariants de ce même sommet. De plus, chaque garde de transition du graphe représente un changement du mode d'exploitation et elle est annotée par une condition

qui décrit les possibles commutations des variables réelles lorsque le système exécute le changement de mode donné.

D'une manière générale, un automate hybride est dit *linéaire* si les conditions de flux, les invariants et les gardes sont définies par des expressions linéaires sur l'ensemble des variables. Dans ce cas, les conditions de flux sont du type  $\dot{x} = constante$  et les invariants et gardes sont du type  $x \sim c$  où  $x$  est une variable,  $c \in \mathbb{Z}$  et  $\sim \in \{<, \leq, =, \geq, >\}$ .

Un exemple d'automate hybride linéaire est donné par la figure 1.5.

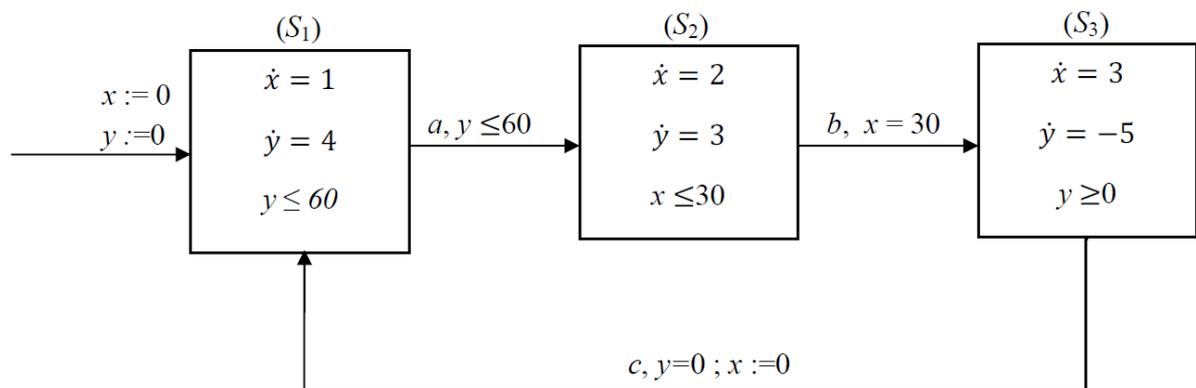


Fig.1.5- Exemple d'automate hybride linéaire.

Dans un automate hybride linéaire, les conditions de flux sont indépendantes des modes de contrôle, et les variables sont indépendantes deux à deux. A chaque franchissement d'une transition de l'automate linéaire, la valeur de chaque variable peut rester inchangée ou bien prendre une nouvelle valeur dans un intervalle donné de possibilités d'une façon non déterministe. Les comportements des variables sont découplés entre elles: les intervalles des valeurs possibles et les valeurs des dérivées pour une variable ne sont pas fonction de la valeur ou de la dérivée d'une autre variable.

### Les automates hybrides rectangulaires

Les *automates hybrides rectangulaires* (Kopke, 1996 ; Henzinger et al., 1998 ; Henzinger et Kopke, 1996, 1999) représentent une plus grande classe que les automates hybrides linéaires.

L'automate hybride rectangulaire se différencie de l'automate hybride linéaire dans le fait que dans chaque mode de contrôle d'un automate rectangulaire, la dérivée première de chaque

variable est donnée dans un intervalle de valeurs possibles et cet intervalle change avec les commutations de contrôle.

Pour résumer, dans un automate hybride rectangulaire, la condition de flux est définie sous la forme de prédicats rectangulaires de la forme  $\dot{x} \in [a, b]$ , pour chaque variable  $x$  du modèle. De même, les invariants, les gardes et la condition initiale sont décrits par des prédicats rectangulaires.

Dans la figure 1.6, un automate hybride rectangulaire est illustré.

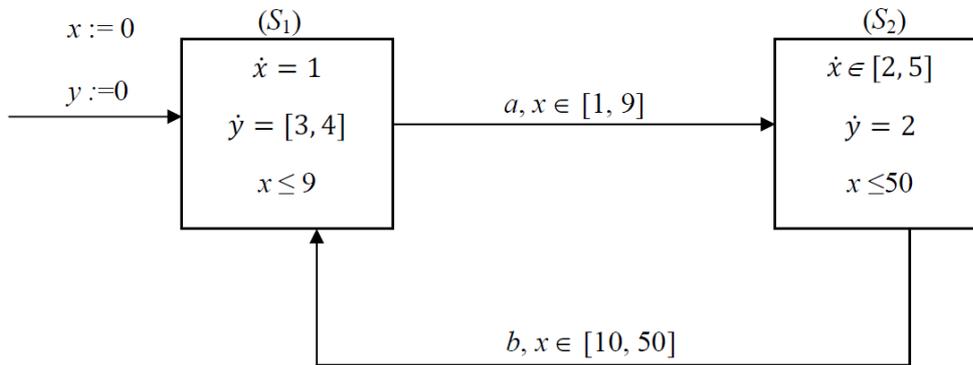


Fig.1.6- Exemple d'automate hybride rectangulaire.

Il existe une sous-classe intéressante des automates hybrides rectangulaires. Il s'agit des *automates hybrides rectangulaires initialisés* (Henzinger et al. 1998).

Dans ce modèle, chaque variable qui change de conditions de flux, suite au franchissement d'une transition entre deux sommets, doit être réinitialisée. Un exemple d'automate hybride rectangulaire initialisé est montré dans la figure 1.7.

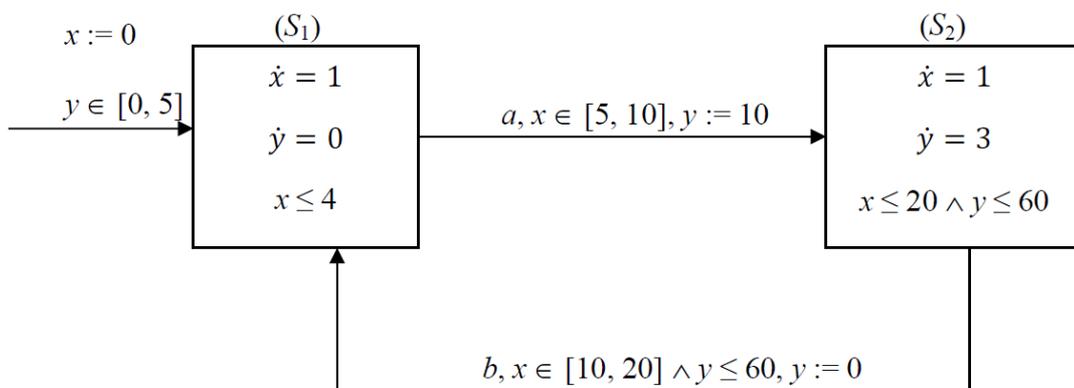


Fig.1.7- Exemple d'automate hybride rectangulaire initialisé.

La sous-classe des automates hybrides rectangulaires représente une sous-classe très intéressante d'automates hybrides. Elle a été largement étudiée et traitée dans la littérature (Spathopoulos, 2000 ; Henzinger et Majumdar, 2000). La pertinence de cette sous-classe revient au fait que de nombreuses questions intéressantes, comme l'analyse d'atteignabilité (Henzinger et al., 1998), la synthèse de contrôleur (Spathopoulos, 2000) et le model-checking (Henzinger et Majumdar, 2000) sont décidables pour les automates hybrides rectangulaires initialisés.

### 1.3.3.3 Réseaux de Petri hybrides

Afin de réaliser la modélisation, l'analyse et la synthèse des systèmes à événements discrets, les réseaux de Petri (Petri, 1962) ont été exploités.

Par la suite, plusieurs efforts ont été mis à contribution dans le but d'adapter ce formalisme pour qu'il puisse traiter les SDH, et une extension des réseaux de Petri (RdP), appelée *réseaux de Petri hybrides*, a été introduite (Le Bail et al, 1991 ; Alla et David, 1998).

Un *RdP hybride* est composé par des places et des transitions continues (C-places et C-transitions) et des places et transitions discrètes (D-places et D-transitions). Le marquage d'une C-place est représenté par un nombre réel, dont l'unité est appelé marque, et le marquage d'une D-place est représenté par un nombre entier dont l'unité est appelé jeton.

En exemple, on modélise un système de fabrication produisant des pièces par lots de cinq par un *RdP hybride*. A la fin de la production de 2 lots de 5 pièces, on entame un nouveau cycle de production. Ce système est illustré par la figure 1.8.

Le marquage de la place  $P_1$  (D-place) est associé au nombre de lots à l'entrée du système de fabrication. Les valeurs  $d_1$  et  $d_2$  représentent les temps de chargement et de déchargement des pièces (D-transitions). La transition  $T_3$  modélise une machine dont la vitesse de production est  $V_3$  (C-transition). Les stocks d'entrée et de sortie de la machine sont respectivement définis par les places  $P_2$  et  $P_3$  (C-places). Le franchissement continu de la transition  $T_3$  correspond à une production continue à la vitesse  $V_3$  quand la place  $P_2$  n'est pas vide. Lorsque  $P_2$  est marqué, le franchissement d'une quantité  $V_3 dt$  de  $T_3$  correspond au retrait de  $V_3 dt$  marques à  $P_2$  et à l'ajout de la même quantité à  $P_3$ .

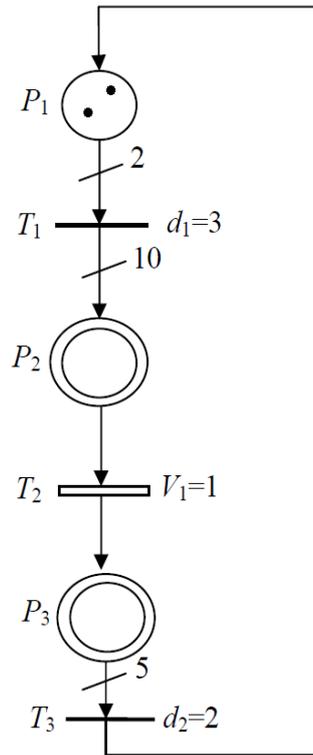


Fig.1.8- Modèle de RdP hybride d'un système de fabrication par lots.

Le modèle RdP hybride possède tous les avantages du modèle RdP de base, comme la représentation du parallélisme, de la synchronisation et des conflits (David et Alla, 2010). Il permet d'obtenir des simulations rapides de systèmes réels. De plus, il est possible de le transformer de manière algorithmique l'automate hybride correspondant (Allam, 1998 ; Allam et Alla, 1996) et d'appliquer tous les outils formels qui y sont développés.

Les RdP hybrides constituent une alternative intéressante dans la modélisation des SDH à dynamique constante pour la commande. Cette idée peut être associée avantageusement à notre travail de recherche.

D'autres approches autour des RdP ont été proposées. On peut citer les travaux de Petterson et Lennartson (Petterson et Lennartson, 1995) qui exploitent les « bond graphs » pour la vérification de systèmes modélisés par des RdP hybrides. L'idée est d'étendre la représentation à des objets événementiels tout en gardant les propriétés des graphes. Giua et Usai (Giua et Usai, 1996) ont introduits, quant à eux, les « High-Level Petri Nets », qui sont caractérisés par l'utilisation de jetons colorés dans la partie discrète du réseau.

### 1.4 Conclusion

Les systèmes dynamiques hybrides sont des systèmes combinant une partie discrète et une partie continue. Ces dernières années, plusieurs travaux se sont intéressés à ce type de systèmes et plusieurs formalismes ont été proposés dans le but d'établir un modèle homogène qui permette la modélisation de cette interaction entre le continu et le discret.

Dans ce chapitre, nous avons présenté dans une première partie les systèmes continus, les systèmes à événements discrets et les systèmes dynamiques hybrides, objet de notre recherche.

Ensuite, nous avons passé en revue les principales approches de modélisation de ces systèmes hybrides. Parmi ces approches, l'approche mixte est celle qui considère les comportements continus et événementiels dans une même structure. L'avantage de cette approche est sa généralité, car elle ne fait pas d'hypothèse sur le type de systèmes à modéliser.

Enfin, nous avons développé quelques outils de modélisation de l'approche mixte. Nous nous sommes particulièrement concentrés sur l'automate hybride et ses différentes sous-classes, introduisant par conséquent l'automate hybride rectangulaire, notre principal outil de modélisation pour la commande des systèmes dynamiques hybrides. Ce dernier sera formellement défini dans le chapitre 3, avant de développer notre approche de commande.

Dans le chapitre suivant, nous allons rappeler les principales approches de commande des systèmes dynamiques autour desquels ce mémoire de thèse est focalisé. Nous axerons notre étude sur les approches de commande issues de la communauté SED. Nous étudierons également quelques extensions de ces approches destinés aux Systèmes Dynamiques Hybrides.



# Chapitre 2

## Commande par supervision

### 2.1 Introduction

Dans le chapitre précédent, nous avons abordé les différents moyens de modélisation des systèmes dynamiques et en particuliers les systèmes dynamiques hybrides. A présent, nous nous intéressons à la commande de ce type de systèmes.

Une partie des approches de la commande des SDH est basée sur le principe de recherche d'une stratégie discrète qui permette de restreindre le comportement du système pour satisfaire certaines spécifications imposées. L'autre partie porte sur la formulation et la résolution d'un problème de commande optimale caractéristique des systèmes continus.

Dans notre cas, on s'intéresse plus à la première partie, car c'est celle qui fait l'objet de notre recherche. Comme ça a été le cas dans le premier chapitre autour de la modélisation, la commande des systèmes hybrides représente une extension de celle des systèmes à événements discrets. Il est alors intéressant d'introduire dans un premier temps les différentes approches de commande des systèmes à événements discrets rencontrées dans la littérature, pour pouvoir par la suite présenter les approches de commande relatives aux systèmes hybrides.

Les premiers travaux sur la commande des systèmes à événements discrets sans le temps ont été introduits par Ramadge et Wonham (Ramadge et Wonham, 1987*b*). Le but de leur approche est de synthétiser un superviseur pour un modèle logique donné (et plus

particulièrement les automates à états finis), de telle sorte que le fonctionnement du procédé combiné avec le superviseur reste toujours inclus dans l'ensemble des comportements valides. De plus, ce fonctionnement doit être le plus permissif possible.

Afin d'enrichir la structure des SED, plusieurs travaux ont étendu cette approche aux cas des SED temporisés (SEDT) en incorporant la notion de temps dans la synthèse de commande. Ceci permet d'avoir une modélisation plus riche en fournissant souvent une solution non vide alors que les méthodes qualitatives sans le temps ont échoué.

Contrairement aux SED et aux SEDT pour lesquels la notion de la commande et les problèmes de synthèse associés sont bien identifiés et clairement définis, la commande des SDH est une notion beaucoup plus large. En effet, le fait que ces systèmes font intervenir deux types de dynamiques, une dynamique continue et une autre discrète, favorise la diversité des formulations du problème de la commande hybride rencontrées dans la littérature. Ainsi, plus la dynamique du système est riche (SED→SEDT→SDH), plus la méthode de synthèse de commande devient complexe.

Dans ce chapitre, nous allons tout d'abord présenter le concept de la commande par supervision. Puis nous allons décrire les différentes approches de commande des SED, des SED temporisés, pour enfin introduire celles des SDH qui nous intéressent dans nos travaux de recherche. A noter que l'accent sera mis sur la synthèse de commande basée sur l'outil automate (automate à états finis, automate temporisé et automate hybride) car il s'agit du modèle utilisé dans notre propre approche de synthèse qui sera développée dans les prochains chapitres.

### **2.2 Commande par supervision : Théorie de base**

La théorie de base de la commande par supervision a été introduite par Ramadge et Wonham (Ramadge et Wonham, 1987b, 1989) pour les systèmes à événements discrets. Dans la littérature, leurs travaux (Ramadge et Wonham, 1987a) sont devenus une référence dans le domaine de la commande à base de modèles des SED. L'étude se place à un niveau qualitatif puisque le temps n'intervient pas. Des modèles logiques (plus particulièrement les automates à états finis) sont utilisés pour décrire le fonctionnement d'un procédé. Le procédé est considéré comme un *générateur d'événements*, c'est-à-dire un SED évoluant de façon

spontanée en générant des événements. On modélise le procédé par un automate  $P=(Q, \Sigma, \delta, q_0)$ . Son comportement est alors donné par son langage  $L(P)$ , qui contient toutes les séquences d'événements qui peuvent être générées par  $P$ .

En général, le comportement du procédé en isolation n'est pas satisfaisant, dans le sens où il ne respecte pas certaines propriétés désirables pour l'exploitation. Ces propriétés représentent les *objectifs de commande* ou *la spécification*. L'objectif de commande revient à restreindre le comportement du procédé afin d'assurer le respect de cette spécification. Un superviseur  $S$  réalise cette restriction. Dans ce contexte, le superviseur se révèle comme une fonction  $S : L(P) \rightarrow 2^\Sigma$  qui, dans chaque état du procédé, lui envoie la liste d'événements interdits par la spécification – *la politique de commande*,  $\Phi$ . Le schéma général de supervision est donné par la figure 2.1.

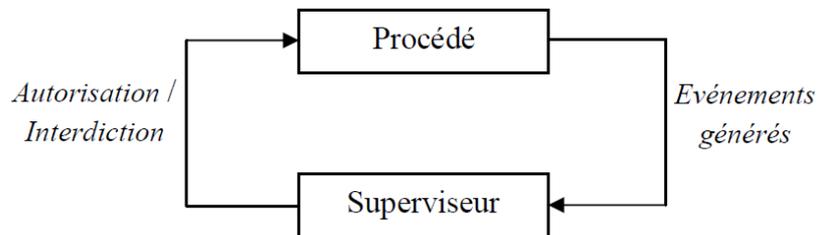


Fig.2.1- Le schéma de supervision.

Le superviseur observe l'état du procédé par l'intermédiaire de la séquence des événements générés par le fonctionnement du système. Cette séquence est l'entrée du superviseur. En réponse, il agit sur le comportement du procédé par l'intermédiaire des lois de commande qui définissent les événements autorisés depuis l'état courant du système.

La théorie de Ramadge et Wonham utilise le modèle automate et les langages formels (Hopcroft et al., 2007) pour modéliser le comportement d'un SED ainsi que les spécifications imposées pour son fonctionnement. En général, un superviseur ne peut pas agir sur tous les événements intervenant dans le fonctionnement d'un procédé. Certains événements ne peuvent pas être interdits. Ce sont les événements incontrôlables. Par conséquent, l'ensemble  $\Sigma$  est partagé en deux sous-ensembles disjoints:  $\Sigma = \Sigma_c \cup \Sigma_u$ , où  $\Sigma_c$  mémorise les événements contrôlables et  $\Sigma_u$  dénote le sous-ensemble des événements incontrôlables. En automatique, les événements contrôlables correspondent en général aux actions appliquées au procédé alors

que les événements incontrôlables sont associés aux sorties (fournies par les capteurs) du procédé.

Une entrée de commande pour un procédé  $P$  est un sous-ensemble  $\gamma \subseteq \Sigma$  tel que  $\Sigma_u \subseteq \gamma$ . Elle correspond à l'ensemble des événements autorisés par le superviseur depuis un état du procédé. Puisque les événements incontrôlables ne peuvent pas être interdits, chaque entrée de commande contient tous les événements incontrôlables. L'ensemble des entrées de commande est noté  $\Gamma$ .

Le fonctionnement du procédé  $P$  non supervisé, modélisé par un langage  $L(P)$ , est appelé fonctionnement en boucle ouverte. Par contre, le fonctionnement du procédé couplé avec son superviseur  $S$ , modélisé par le langage  $L(S/P)$ , est appelé fonctionnement en boucle fermée. Ce fonctionnement doit respecter les spécifications imposées par le cahier des charges. Comme nous l'avons mentionné précédemment, la tâche d'un superviseur est de générer l'entrée de commande de telle sorte que le procédé évolue avec un maximum de liberté tout en respectant certaines contraintes. Cependant, à cause de l'existence des événements incontrôlables, il n'est pas possible de restreindre le comportement du procédé à n'importe quel sous-ensemble.

Un langage désiré d'un procédé est l'ensemble des mots générés par le procédé tout en respectant les contraintes imposées par le cahier des charges. Soit  $K$  le langage désiré pour un procédé  $P$ . L'étude de l'existence d'un superviseur  $S$  tel que  $L(S/P) = K$  est basée sur la propriété de contrôlabilité du langage  $K$ .

Dans la pratique, la synthèse du superviseur est effectuée à l'aide de l'outil automate. Le comportement du procédé à superviser ainsi que les spécifications imposées pour son fonctionnement sont modélisés par des automates. Le langage désiré du procédé est représenté par l'automate obtenu par la composition synchrone des automates qui modélisent le procédé et les spécifications.

D'une manière générale, la propriété de contrôlabilité du langage désiré d'un SED n'est pas nécessairement vérifiée a priori. Par conséquent, la construction d'un superviseur commence par la recherche du plus grand sous-langage contrôlable inclus dans le langage désiré pour le procédé. Dans leur approche, Ramadge et Wonham proposent un algorithme itératif pour la construction du plus grand sous-langage contrôlable inclus dans un langage donné.

Un point important à aborder lors de la discussion sur la commande est celui de l'optimalité. Deux aspects importants permettent d'assurer l'optimalité :

- L'interdiction des comportements indésirables (illustrée dans le schéma de supervision)
- La garantie que le système n'est soumis à aucune restriction inutile. Dans ce cas, il s'agit de la *condition de comportement maximal permissif*.

L'approche de Ramadge et Wonham (Ramadge et Wonham, 1987a) assure l'optimalité du contrôleur pour les modèles automates à états finis. Notre approche de commande des systèmes hybrides par automates hybrides rectangulaires, qu'on traitera dans les chapitres suivants, assurera aussi la condition de commande maximale permissive.

**Remarque 2.1.** Dans notre démarche, on traite davantage de la complexité de la modélisation plutôt que de la notion de contrôlabilité. C'est pour cela que tous les événements qu'on traitera seront contrôlables.

□

### Exemple 2.1. Tronçons de circulation

Dans ce qui suit, nous introduisons un système de deux tronçons de circulation que nous allons traiter tout au long de ce mémoire. Ce système est illustré dans la figure 2.2.

Il s'agit de deux tronçons de circulation  $T_1$  et  $T_2$  où le trafic routier est régulé par un policier (ou un feu de circulation). Chaque tronçon est modélisé par un automate à 2 états : arrêt ( $q_0$  et  $q_2$ ) et marche ( $q_1$  et  $q_3$ ). Dans l'état initial, le trafic est à l'arrêt. Le modèle du procédé est obtenu en construisant le composé synchrone de  $T_1$  et  $T_2$  :  $P = T_1 \parallel_s T_2$  (figure 2.3). Le fonctionnement de  $P$  est décrit sur l'alphabet  $\Sigma = \{Stop_1, Start_1\} \cup \{Stop_2, Start_2\}$  où  $\{Stop_1, Start_1\}$  (resp.  $\{Stop_2, Start_2\}$ ) est l'alphabet du tronçon  $T_1$  (resp.  $T_2$ ). Tous les événements considérés dans cet exemple sont contrôlables (Remarque 2.1). Cette hypothèse peut être levée ici. Cependant, on la conserve car elle sera prise en compte dans le modèle hybride.

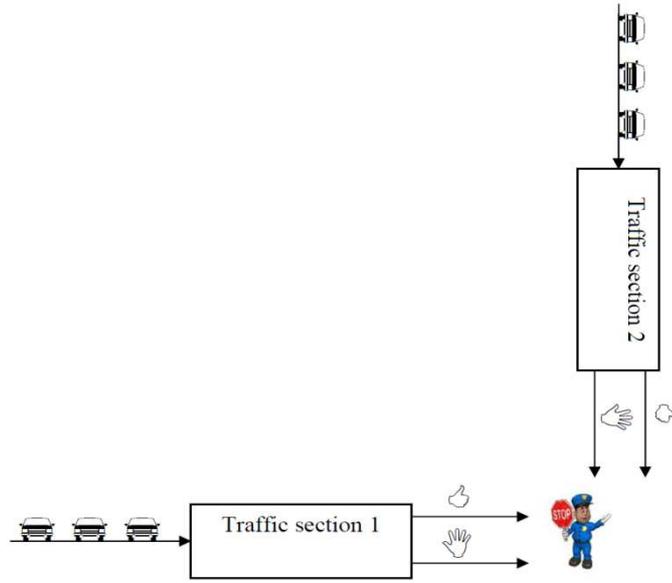


Fig.2.2- Le système de circulation.

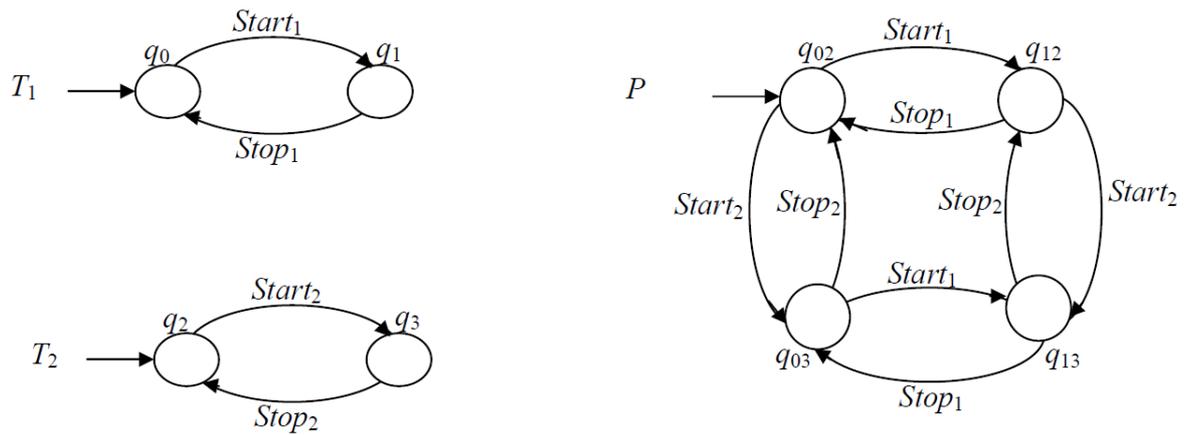


Fig.2.3- Tronçons à 2 états et procédé en isolation.

La contrainte de circulation (figure 2.4) est la suivante : le policier autorise successivement le flux dans  $T_1$ , l'arrête, lance le flux dans  $T_2$  et l'arrête.

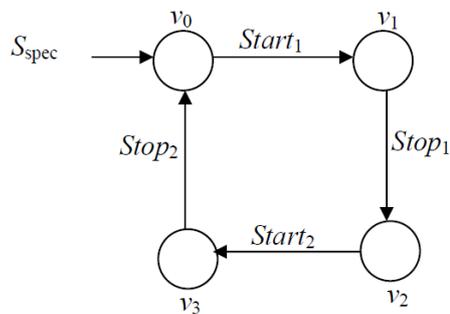


Fig.2.4- Spécification.

Le système supervisé (boucle fermée) est le composé synchrone des modèles du procédé et de la spécification :  $S/P = P \parallel_s S_{\text{spec}}$ . L'accepteur  $S/P$  reconnaît le fonctionnement désiré  $L(S/P) = L(P) \cap L(S_{\text{spec}})$ . Puisque tous les événements sont contrôlables, le modèle du contrôleur est donné dans la figure 2.5.

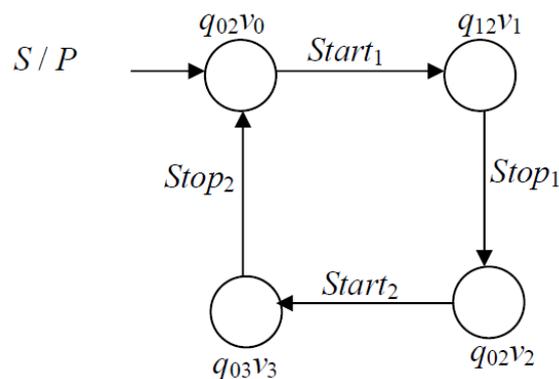


Fig.2.5- Modèle du contrôleur pour le système tronçon de circulation.

Même si cette synthèse peut paraître triviale, elle n'est présentée que pour montrer la démarche progressive de notre travail. Tout ce que l'on peut faire sur ce modèle c'est gérer le séquençement des feux. La structure utilisée n'est pas riche. On ne peut ni gérer la durée des feux ni le flux des véhicules. Dans la suite, nous allons voir comment l'ajout d'une contrainte temporelle dans un premier temps puis de la dynamique des variables dans un deuxième temps va enrichir le modèle automate et par conséquent la commande par supervision.

### Avantages et inconvénients de la théorie de Ramadge et Wonham

Comme nous venons de le voir dans l'exemple et de manière générale, la théorie de Ramadge et Wonham représente un centre d'intérêt considérable pour la communauté scientifique concernée par ce domaine. Plusieurs groupes de recherche s'intéressent, sur le plan international, à l'extension et l'optimisation de cette approche. Celle-ci utilise les automates à états finis et permet d'obtenir de bons résultats. Cependant, son application à des systèmes complexes peut devenir difficile, voir impossible à cause du problème de l'explosion combinatoire d'états. Il y a deux manières de compenser :

- soit en choisissant des modèles différents comme les RdP Petri (Yamalidou et al., 1996 ; Moody et Antsaklis, 1998 ; 2000 ; Giua et al., 1992 ; Badouel et al., 1994 ; Ghaffari et al., 2003 ; Iodarche et al., 2001 ; Wu et Zhou, 2004 ; Holloway et Krogh, 1990 ; Vasiliu, 2012)

- soit en ajoutant des dynamiques plus riches comme le temps (Gaubert, 1995 ; Brandin et Wonham, 1994 ; Sava, 2002), et les flux des variables (Sreenivas et Krogh, 1992 ; Uzam et Wonham, 2006).

### **2.3 Commande à base de modèles temporisés des Systèmes à Événements Discrets**

Jusqu'à présent, nous nous sommes intéressés à la supervision et la commande par supervision des SED à base de modèles qui ne prennent en considération que l'ordre d'occurrence des événements, et ne tiennent pas compte de la date d'occurrence de l'événement. Cependant, de nombreux travaux ont introduit la notion de temps dans la synthèse de commande. Ceci permet d'avoir une modélisation plus riche en fournissant souvent une solution non vide alors que les méthodes qualitatives sans le temps ont échoué. Ainsi par exemple il est possible de prendre en compte des systèmes tels que la fin d'une tâche a lieu dans un intervalle dont la largeur exprime l'incertitude sur la durée de l'opération. L'information temporelle peut être utilisée pour synthétiser des lois de commande moins contraignantes dans le sens où l'on peut restreindre la contrainte temporelle sur l'occurrence d'un événement sans l'interdire totalement.

Plusieurs travaux basés sur différents outils de modélisation ont été élaborés pour l'analyse et la commande des systèmes à événements discrets temporisés (SEDT). Dans la suite, nous présentons quelques unes de ces approches, classées en deux catégories : les méthodes en temps discret et les méthodes en temps continu.

#### **2.3.1 Commande par supervision en temps discret**

La théorie de Ramadge et Wonham pour la commande des SED présentée dans la section précédente a été étendue par Brandin et Wonham (Brandin et Wonham, 1994) en vue de la commande des SEDT. Dans leurs travaux, ils ont discrétisé le temps pour se ramener à une étude basée sur l'approche de Ramadge et Wonham. C'est là qu'est apparu la notion d'événement forcé.

Deux types d'automates sont utilisés dans cette approche, le premier pour modéliser les états et les transitions entre les états d'un système et le deuxième pour modéliser les

contraintes temporelles dans le modèle du comportement du SED. Comme pour les SED, les événements peuvent être soit contrôlables soit incontrôlables. Par conséquent, l'ensemble des événements est composé de trois sous-ensembles disjoints :

$$\Sigma = \Sigma_c \cup \Sigma_u \cup \{tick\}$$

Où :

- $\Sigma_c$  est l'ensemble des événements contrôlables ;
- $\Sigma_u$  est l'ensemble des événements incontrôlables.
- *tick* est l'événement représentant l'occurrence d'un top d'horloge

Il existe une autre catégorie d'événements, qui est essentielle dans la commande supervisée des SEDT. Il s'agit des événements forçables. Un événement est dit forçable s'il peut se produire spontanément ou être forcé par un système extérieur tout en respectant la contrainte temporelle associée à sa date d'occurrence. Dans l'approche de Brandin et Wonham, un événement forçable peut préempter l'occurrence de l'événement *tick*. Par conséquent, le superviseur peut forcer l'occurrence d'un événement avant que l'horloge atteigne une certaine valeur. On note par  $\Sigma_{for}$  l'ensemble des événements forçables. Il n'existe aucune relation entre l'ensemble des événements forçables et les ensembles des événements contrôlables et incontrôlables.

Considérons un procédé  $P$  à superviser. Son comportement temporisé est modélisé par un automate  $A_P = (Q, \Sigma, \delta, q_0, Q_m)$  générant un langage  $L(A_P)$ . Soit un mot  $w \in L(A_P)$ . Il existe alors un état  $q \in Q$  atteint par l'exécution du mot  $w$  à partir de l'état initial  $q_0$ . On décrit les possibilités d'évolution du procédé depuis cet état par l'ensemble d'événements éligibles dans l'état  $q$ .

Comme pour la théorie de supervision des SED, les événements incontrôlables sont toujours autorisés par le superviseur. Cependant, si parmi les événements éligibles il y a au moins un événement forçable, le superviseur peut forcer son exécution avant l'occurrence d'un nouveau top d'horloge (événement *tick*). Dans cette approche le superviseur joue aussi le rôle d'un système de commande.

L'étude du comportement en boucle fermée, du concept de contrôlabilité d'un langage ainsi que le calcul du langage suprême contrôlable sont abordés de la même façon que dans

la théorie de Ramadge et Wonham. On montre en plus que le superviseur trouvé est le plus permissif.

### Avantages et inconvénients de l'approche de Brandin et Wonham

L'approche de Brandin et Wonham donne une bonne solution de commande par supervision des SEDT. La matérialisation de l'écoulement du temps à travers un événement spécifique inclut la dimension temporelle au sein même d'un langage comparable au langage d'un automate non temporisé.

Cependant, cette méthode a un inconvénient majeur au niveau de la modélisation. On retrouve le problème de l'explosion combinatoire du nombre d'états pour les modèles SED. D'autre part, l'aspect discret du temps est une approximation dans la modélisation du système.

### 2.3.2 Commande par supervision en temps continu

Afin de remédier au problème de l'explosion combinatoire engendrée par une nature discrète du temps, plusieurs approches de commande par supervision ont été proposées. Principalement basées sur l'outil automate temporisé (Alur et Dill, 1994), ces approches considèrent que le temps évolue d'une façon continue.

On peut illustrer cette idée par l'exemple précédent des deux tronçons de circulation en y intégrant la notion de temps. On peut alors modéliser chaque tronçon par un automate temporisé comme le montre la figure 2.6. L'état d'un automate temporisé est l'association d'un sommet de l'automate (état discret) avec l'ensemble de valeurs réelles des horloges à un instant donné (état continu). On a une horloge  $x$  associée à chaque sommet, qui mesure la durée des feux : « Rouge » dans  $q_0$  (resp.  $q_2$ ) et « Vert » dans  $q_1$  (resp.  $q_3$ ).

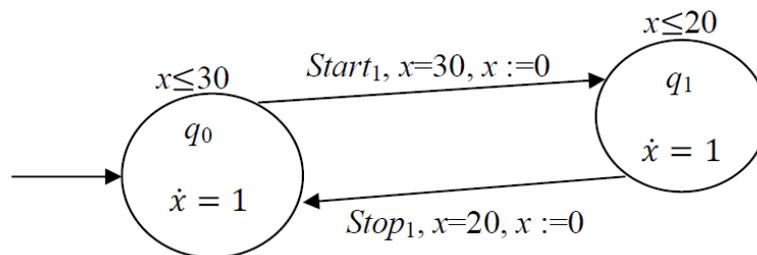


Fig.2.6- Automate temporisé possible d'un tronçon de circulation.

On remarque bien que l'information temporelle enrichit l'automate. Concernant la spécification, elle peut maintenant porter sur l'événement et/ou la durée des feux, ce qui rend la synthèse de commande plus complexe mais plus riche. La figure 2.7 donne un exemple de spécification : on veut que la durée des feux dans les deux tronçons soit la même et égale à 40 u.t.

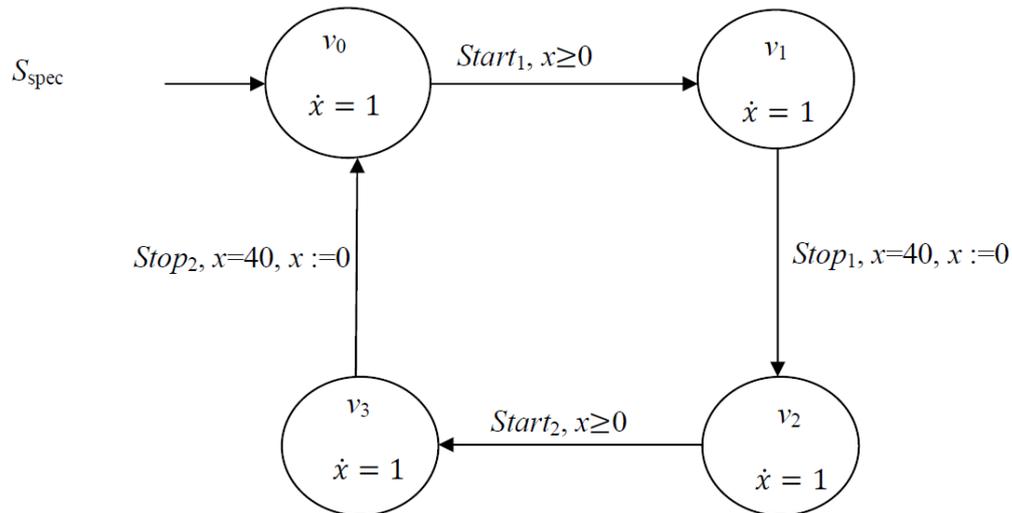


Fig.2.7- Spécification possible pour les tronçons de circulation.

**Remarque 2.2.** Le modèle temporisé étant une sous-classe du modèle hybride, cet exemple sera traité par notre approche. □

Dans la suite, on présente quelques unes des approches de commande en temps continu des SEDT.

### 2.3.2.1 Commande par supervision basée sur l'automate temporisé à retards

Cette approche introduite dans (Altisen et al., 1999) utilise le modèle réseau de Petri à retards pour la modélisation du processus à commander. La synthèse de commande par supervision, quant à elle, est basée sur l'outil automate temporisé à retards.

Le réseau de Petri à retards possède la structure discrète du modèle réseau de Petri autonome (David et Alla, 1989 ; Murata, 1989), alors que l'information temporelle est modélisée comme dans le modèle automate temporisé (Yovine, 1993 ; Olivero, 1994). Le réseau de Petri à retards est utilisé seulement pour modéliser le procédé, sans prise en compte des spécifications imposées sur son fonctionnement.

L'étape qui suit consiste à modéliser le comportement du réseau de Petri à retards par un automate temporisé à retards. Ce dernier correspond à un automate temporisé dans lequel on remplace les invariants des sommets par des conditions de franchissement au plus tard associées aux transitions. On exprime la spécification imposée sur le fonctionnement du procédé par une propriété qui doit être satisfaite par les états de l'automate temporisé à retards correspondant au procédé. On considère un automate temporisé  $A$  et une propriété  $S$ . Le but de la synthèse d'un superviseur est de construire un automate à retards  $A_S$  tel que tous ses états vérifient la propriété  $S$ .

On note par  $Q$  le sous-ensemble des états de l'automate  $A$  satisfaisant la propriété  $S$ . L'approche de synthèse, développée dans (Maler et al., 1995), permet de déterminer l'ensemble  $\pi(Q)$  des états à partir desquels on peut atteindre n'importe quel état  $q \in Q$  par le franchissement d'une transition contrôlable. On retire ainsi de l'ensemble  $\pi(Q)$  les états à partir desquels on peut évoluer, par le franchissement d'une transition incontrôlable, vers un état n'appartenant pas au sous-ensemble  $Q$ .

Le calcul du superviseur se fait d'une manière itérative. La procédure commence par l'initialisation  $Q[0] = Q$ . A chaque itération  $i$  on calcule l'ensemble  $\pi(Q[i])$  et on réactualise  $Q[i+1] = Q[i] \cap \pi(i)$ . L'algorithme s'arrête lorsque  $Q[i+1] = Q[i]$ . On note par  $Q^*$  l'ensemble obtenu. Cette procédure de synthèse de la supervision mène à un automate  $A_S$  ayant la même structure discrète que l'automate  $A$  sauf pour les gardes des arcs contrôlables qui sont modifiées de telle sorte que seuls les états  $q \in Q^*$  soient atteignables.

Grâce à cette approche, on peut déterminer toutes les séquences de transitions contrôlables de telle sorte que le système n'évolue pas vers des états ne respectant pas la propriété  $S$ .

### **Exemple 2.3. Document multimedia (Altisen et al., 1999)**

Les auteurs proposent plusieurs applications de l'algorithme de commande, dont l'exemple du document multimédia composé des activités basiques suivantes avec leur durée [minimale, maximale]: musique [30, 40], vidéo [15, 20], audio [20, 30], texte [5, 10], application [20, 30], image [20,  $\infty$ ]. Au départ, les activités musique, vidéo, audio et application sont lancées en parallèle. Elles sont soumises aux contraintes de synchronisation suivantes :

- 1- Video et audio terminent dès que l'un d'entre eux finit. Leur terminaison est suivie immédiatement par l'affichage du texte.
- 2- Musique et texte doivent terminer en même temps
- 3- L'application est suivie par une image
- 4- Le document termine dès que l'image et la musique (et le texte) ont fini
- 5- Les temps d'exécution de l'audio et de l'application dépendent du chargement de la machine et sont donc incontrôlables.

Le réseau de Petri à retards modélisant le système est donné par la figure 2.8.

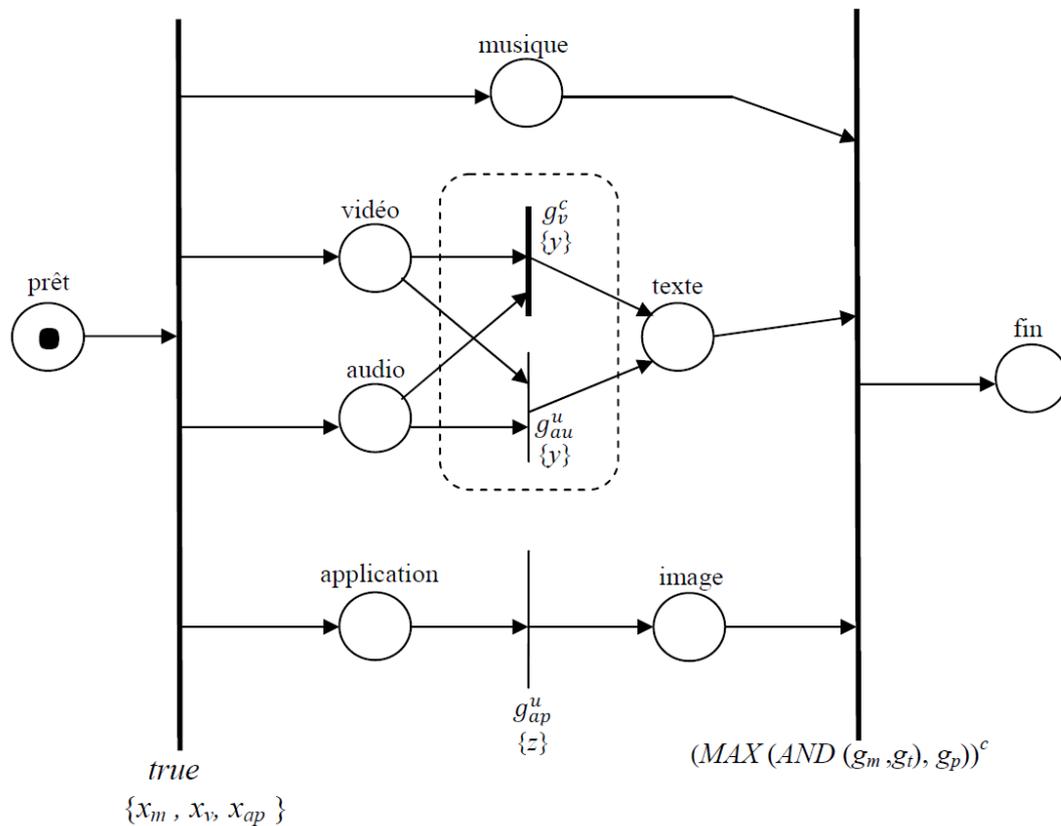


Fig.2.8- RdP à retards du système multimedia.

L'automate temporisé obtenu à partir du réseau de Petri est donné par la figure 2.9. Les transitions contrôlables sont en traits épais et notée par la lettre « c ». Les transitions incontrôlables sont en traits fins et notée par la lettre « u ». Les gardes de transitions sont comme suit :  $g_m=(30 \leq x \leq 40)$ ,  $g_v=(15 \leq y \leq 20)$ ,  $g_{au}=(20 \leq y \leq 30)$ ,  $g_t=(5 \leq y \leq 10)$ ,  $g_{ap}=(20 \leq z \leq 30)$ ,  $g_i=(20 \leq z)$ .

On applique l'algorithme de synthèse pour des propriétés d'atteignabilité sur cet exemple. On cherche dans ce cas l'existence d'un superviseur qui amène le système de l'état initial vers l'état « *fin* ». Le superviseur obtenu est donné par la figure 2.10. Les gardes restreintes des transitions contrôlables calculées par l'algorithme de synthèse sont en gras.

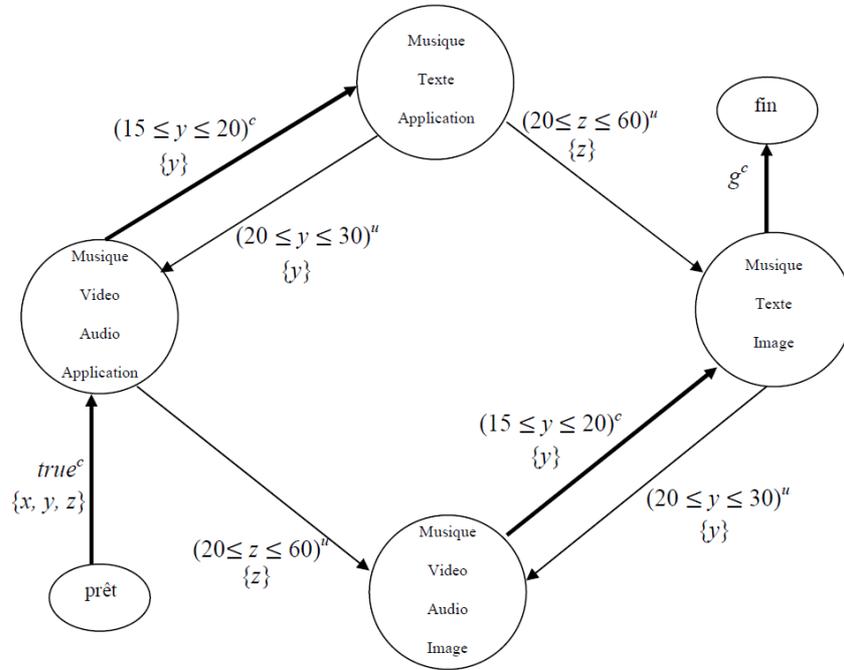


Fig.2.9- Automate temporisé à retards du document multimédia.

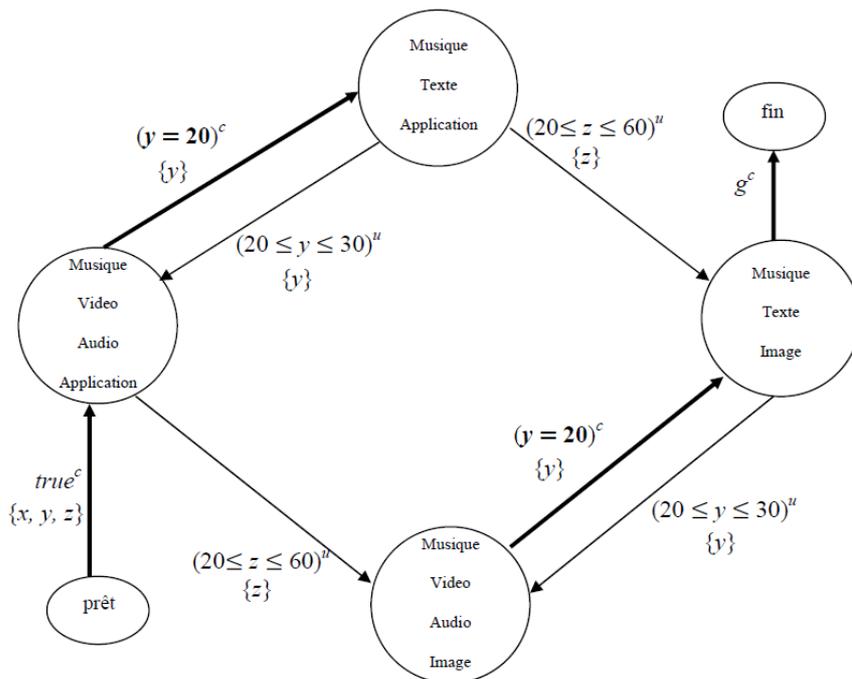


Fig.2.10- Superviseur multimédia.

### 2.3.2.2 Commande par supervision basée sur l'automate à temps continu

Cette approche a été présentée dans (Sava, 2002). Elle propose l'utilisation du modèle réseau de Petri à arcs temporels pour la modélisation du procédé et des spécifications désirées. Quant à la synthèse de la commande par supervision, elle est basée sur l'outil automate à temps continu. Le RdP à arcs temporels hérite de la capacité de modélisation des modèles RdP. Cependant, il est moins approprié pour l'analyse.

Afin de réaliser la synthèse de la commande, on construit l'automate à temps continu modélisant le comportement du RdP à arcs temporels. Chaque état du RdP à arcs temporels est modélisé par un sommet de l'automate à temps continu. Par conséquent, les états interdits sont modélisés par des sommets interdits. Chaque franchissement d'une transition du RdP est modélisé par le franchissement d'une transition de l'automate. Les transitions d'un automate à temps continu peuvent être soit contrôlables soit incontrôlables selon la nature de la transition correspondante dans le modèle RdP à arcs temporels. Les contraintes temporelles sur la date de franchissement des transitions d'un RdP à arcs temporels sont modélisées par des gardes associées aux transitions de l'automate.

La procédure de synthèse de commande par supervision consiste à analyser un par un tous les sommets interdits et à restreindre les gardes des transitions contrôlables telles que ces sommets ne soient plus jamais atteignables. Cette méthode est basée sur une technique de calcul du temps minimal et maximal de séjour du système dans un sommet de l'automate.

**Exemple 2.4.** Soit la partie de l'automate à temps continu présentée dans la figure 2.11.

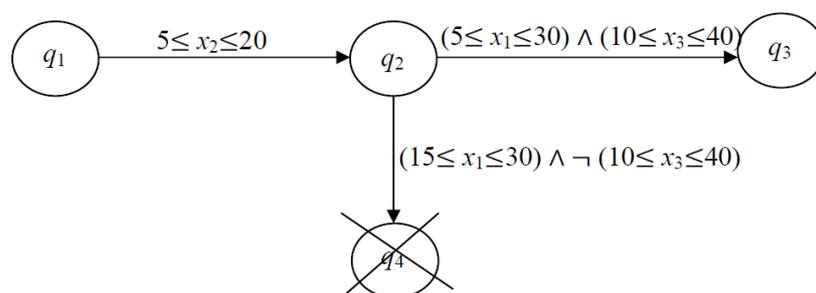


Fig.2.11- Portion d'un automate à temps continu.

Le sommet interdit  $q_4$  est atteint à partir du sommet  $q_2$ . Le but de la synthèse est de garantir qu'à partir du sommet  $q_2$ , la transition vers  $q_3$  est toujours franchie avant celle vers  $q_4$ . On

cherche alors des nouvelles valeurs pour les gardes de transitions contrôlables d'entrée et de sortie de  $q_2$  garantissant que le temps maximal de séjour dans  $q_2$  avant d'évoluer vers  $q_3$  soit plus petit que le temps minimal de séjour de  $q_2$  avant d'évoluer vers  $q_4$ .

Cette approche est basée sur deux procédures :

- Le *contrôle aval* : cette procédure consiste à modifier la garde de la transition de sortie de  $q_2$  vers  $q_3$  telle que cette transition soit toujours franchie avant celle vers le sommet interdit. Elle ne peut être appliquée que si la transition de  $q_2$  vers  $q_3$  est contrôlable.
- Le *contrôle amont* : cette procédure consiste à modifier la garde de la transition d'entrée dans  $q_2$  telle que la valeur des horloges dans ce sommet ne permette pas de franchir la transition vers  $q_4$ . Elle ne peut être appliquée que pour les transitions d'entrée contrôlables.

Si la procédure avale ne permet pas de trouver une solution, le sommet  $q_2$  devient interdit. Ce processus de calcul doit être répété jusqu'à convergence.

Pour conclure, cette approche de synthèse associe la capacité de modélisation de l'outil RdP à arcs temporels avec la puissance d'analyse du modèle automate à temps continu. Les avantages de cette démarche sont d'une part le fait qu'elle permette d'obtenir un modèle clair et concis du procédé étudié et d'autre part le fait d'éviter l'apparition d'incohérences temporelles. L'inconvénient de cette approche est qu'elle a mené à une résolution prouvée du problème local de la synthèse de commande et que le problème global reste ouvert.

### 2.3.2.3 Commande par supervision basée sur l'automate temporisé de jeux

L'utilisation de la théorie des jeux dans le but de réaliser la synthèse de contrôleurs pour les systèmes à événements discrets temporisés modélisés par des automates temporisés (Alur et Dill, 1994) a été décrite dans (Altisen et al, 2005 ; Cassez et Markey 2007).

Le *problème de contrôle* (CP) a été formulé dans le cadre de la *théorie des jeux* temporisés dans (Maler et al., 1995 ; Asarin et al. 1998 ; Faëlla et al., 2002 ; De Alfaro et al., 2003). Le système est modélisé par un *automate temporisé de jeux*. Il peut contenir des transitions contrôlables et incontrôlables. Les actions de transition peuvent alors être classées en deux groupes,  $\Sigma_c$  pour les actions contrôlables et  $\Sigma_u$  pour les actions incontrôlables. Les protagonistes du jeu sont le contrôleur (qui détermine quand une action de  $\Sigma_c$  est faite) et

l'environnement (qui peut faire des actions de  $\Sigma_i$ ). Lors du déroulement du jeu, on doit définir la marche à suivre et les coups possibles pour chaque joueur. Concernant les jeux discrets, les coups possibles sont des actions discrètes. Pour les jeux temporisés (Maler et al., 1995 ; Asarin et al. 1998), les joueurs disposent d'une action supplémentaire qui est l'attente : un joueur peut décider d'attendre avant de faire une action discrète si celle-ci ne doit pas être jouée trop tôt. Dans les deux cas, la légalité et le résultat d'une action sont déterminés par les *règles du jeu*.

Le temps (quantitatif) dans le modèle a pour intérêt de décrire des joueurs dont les comportements dépendent du temps, et leurs actions sont donc fonctions des délais écoulés. Il existe deux *sémantiques* classiques pour les jeux temporisés : les jeux à *observation continue* (Maler et al., 1995 ; Asarin et al., 1998) et Les jeux avec *surprise* (De Alfaro et al., 2003).

Le CP s'écrit dans le cadre de la théorie des jeux de la manière suivante : « Existe-t-il une *stratégie* telle que le contrôleur gagne le jeu ? ». La stratégie montre le coup à jouer, en fonction de l'historique des coups joués par les joueurs et des états rencontrés depuis le début du jeu.

Les objectifs de contrôle les plus simples, donnés par la propriété souhaitée  $\phi$ , sont des objectifs de *sûreté* et d'*atteignabilité*. D'une manière plus générale, les objectifs de contrôle peuvent être définis comme étant des formules de logiques temporelles (par exemple LTL) (Pnueli et Rosner, 1989). On peut spécifier des objectifs comme « amener infiniment souvent le jeu dans  $\phi_1$  sans jamais passer par  $\phi_2$  ». De tels objectifs sont dits  $\omega$ -réguliers (Thomas, 1995).

Le CP se ramène à un problème de calcul des états *gagnants* du jeu. Ces derniers sont définis sémantiquement comme étant les états à partir desquels il existe une stratégie gagnante.

Plusieurs algorithmes de synthèse de contrôleurs pour les automates temporisés de jeu ont été proposés (Maler et al., 1995 ; Asarin et al. 1998 ; De Alfaro et al., 2001 ; De Alfaro et al., 2003). Des travaux récents ont proposé des implémentations efficaces de l'algorithme proposé dans (Maler et al., 1995) (Altisen et Tripakis, 1999 ; 2002 ; Cassez et al. 2005). Puis, les algorithmes de synthèse de contrôleurs ont été étendus aux cas des automates hybrides (Henzinger et al., 1999 ; Henzinger et Kopke, 1999 ; Wong-Toi, 1997), généralisation des automates temporisés.

## 2.4 Commande des Systèmes Dynamiques Hybrides

Le fait que les systèmes hybrides fassent intervenir une dynamique continue et une dynamique discrète rend la notion de commande beaucoup plus large. Par conséquent, plusieurs formulations du problème de la commande hybride existent dans la littérature. Parmi les méthodes qui traitent explicitement de la commande des systèmes hybrides, certaines accordent une importance plus grande à la partie discrète, et d'autres s'intéressent davantage à la dynamique continue. Notre travail de recherche se plaçant dans la première catégorie, nous allons, dans la suite, mettre l'accent sur les approches événementielles. Une présentation brève des méthodes à dynamique continue sera faite dans une première partie.

### 2.4.1 Commande continue des Systèmes Dynamiques Hybrides

Certaines approches de commande des Systèmes Hybrides s'intéressent à la dynamique continue. Dans ce cas, la classe envisagée de ces systèmes est celle des systèmes continus avec commutation de modèle. La commande recherchée est hybride (commande continue et discrète) et les spécifications portent essentiellement sur la partie continue (régulation, optimisation d'un critère portant sur les variables continues, etc.). On trouve ainsi, parmi ces approches, des méthodes fondées sur la formulation et la résolution d'un problème de commande optimale, où le critère à minimiser porte uniquement sur les variables continues, et des méthodes basées sur la théorie de Lyapunov (Zaytoon, 2001). Il existe une multitude d'approches, dans ce qui suit, on va n'en citer que quelques unes.

Branicky et Mitter (Branicky et Mitter, 1995) ont proposé une première approche de commande optimale des SDH. Se basant sur son modèle unifié (Branicky, 1998), Branicky introduit des outils d'analyse appliqués aux systèmes à commutation et hybrides. L'auteur présente particulièrement les « fonctions multiples de Lyapunov » comme un outil d'analyse pour la stabilité des systèmes à commutation. L'idée est d'imposer des restrictions sur la commutation pour garantir la stabilité, même si à chaque système individuel est associée une fonction Lyapunov.

Une autre méthode de commande optimale des SDH a été proposée dans (Lygeros et al., 1996). Il s'agit d'une approche pour la conception de systèmes multi-agents basée sur la commande optimale et la théorie des jeux. Dans ce cas, le système est perçu comme étant un jeu entre deux joueurs, le contrôleur et la perturbation, qui concourent sur des fonctions de

coût (lesquelles ont un rapport avec les propriétés que le système en boucle fermée doit satisfaire). Le contrôleur gagne le jeu s'il peut maintenir le système dans un fonctionnement « sûr » en présence de perturbations. Cette approche a été appliquée dans (Lygeros et al., 1998) pour la commande des systèmes type AHS (Automated Highway Systems) et utilisée dans (Tomlin et al., 1998) pour la gestion du trafic aérien. L'inconvénient des semi-algorithmes proposés dans ces approches (Lygeros et al., 1999 ; Tomlin, 1998), c'est que leur terminaison n'est pas garantie.

### 2.4.2 Commande événementielle des Systèmes Dynamiques Hybrides

A présent, on s'intéresse aux méthodes de commande qui accordent une importance plus grande à la partie discrète. Dans ce cas, les systèmes hybrides concernés sont généralement modélisés sous la forme d'automates hybrides. Le problème de commande correspondant est formulé sous la forme de recherche d'une stratégie discrète permettant de restreindre la fonction de transition du système pour satisfaire les spécifications (Alur et al., 1995 ; Asarin et Maler, 1999).

L'état du système non commandé est discret/continu, c'est-à-dire qu'il est défini par un état discret (une situation à laquelle sera associé l'état des commandes du système) et un état continu (état de sorties du système : température, concentration, niveau, etc.). L'objectif est de synthétiser une commande événementielle permettant de piloter le système à partir d'une région hybride initiale tout en respectant des contraintes de fonctionnement exprimées en termes de valeurs limites sur les variables continues et de séquençement d'états discrets autorisés, etc.

Dans la suite, nous allons présenter quelques unes des approches de commande événementielle.

Tittus (Tittus, 1995 ; Tittus et Edgart, 1998) étudie la synthèse de contrôleurs pour une classe de systèmes hybrides où la dynamique continue est décrite par des intégrateurs en utilisant le modèle de l'automate hybride linéaire. Afin que l'évolution d'un système de dimension quelconque puisse être contrôlée depuis une région hybride initiale jusqu'à une région hybride finale, il est impératif que le système soit contrôlable, au sens de Tittus : un système est contrôlable si et seulement s'il existe un chemin fini (système non zénon) reliant toute situation initiale à une situation finale en un temps fini, en respectant les fonctions

d'évolution et les invariants de chaque situation. Par conséquent, la contrôlabilité est relative à des régions et non à des points. La recherche de contrôlabilité (ou d'atteignabilité) est faite par inférence arrière de l'état d'arrivée vers l'état de départ, comme en programmation dynamique. En général, ce problème n'est pas décidable. Cependant, Tittus (Tittus, 1995 ; Tittus et Edgart, 1998) a développé des algorithmes pour des systèmes intégrateurs. Ces algorithmes ont par la suite été étendus à des systèmes à champs de vecteurs linéaires (Manon et Valentin-Roubinet, 1999a, 1999b). La loi de commande générée à partir de l'étude de contrôlabilité correspond à la stratégie de commutation au plus tôt.

L'approche d'Antsaklis (Antsaklis et al., 1993 ; Stiver et Antsaklis, 1993 ; Koutsoukos et al., 2000) a été à l'origine de nombreux travaux dans le domaine des SDH. Il s'agit d'une approche de synthèse d'un superviseur discret pour un système continu, qui est basée sur une adaptation de la méthode de Ramadge et Wonham. Le procédé continu est décrit par un ensemble d'équations différentielles, alors que le contrôleur discret est modélisé par un automate à états finis. Par conséquent, une interface est définie entre la partie SED et la partie continue. D'un côté, les variables continues du procédé sont transformées en symboles par une fonction appelée « générateur ». De l'autre, une fonction appelée « actionneur » transforme les événements provenant du superviseur en signaux continus par morceaux utilisés en tant qu'entrée du procédé. On illustre cette approche par la figure 2.12.

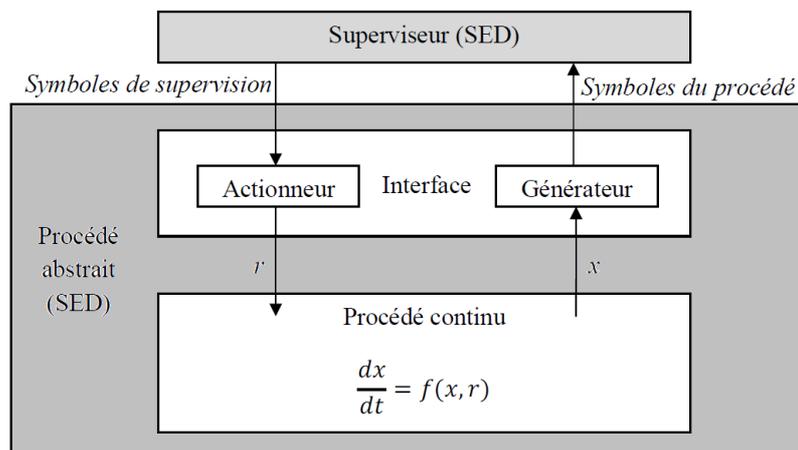


Fig.2.12- Approche d'Antsaklis.

Dans la méthode d'Antsaklis, l'objectif est de respecter une spécification. On peut citer comme type de spécifications :

- Objectif de sécurité : ne jamais atteindre une situation de l'automate

- Atteignabilité : être toujours capable de retourner à une situation de l'automate
- Appartenance des trajectoires du procédé abstrait à un ensemble donné (les mêmes séquences de symboles du procédé se répètent périodiquement, à l'exclusion de toutes autres).

La méthode d'Antsaklis possède l'avantage d'être proche de la pratique industrielle qui utilise un automate programmable industriel pour commander le procédé continu à travers des capteurs et des actionneurs tout ou rien. De ce fait, cette méthode peut fournir un support formel à ce type d'automatisation industrielle, qui est généralement basée sur une approche intuitive de synthèse de commande.

Dans (Pettersson et Lennartson, 1996), les auteurs proposent une méthode analytique de conception d'un contrôleur conduisant un système hybride intégrateur ( $\dot{x} = \gamma_j$  où  $\gamma_j$  sont des constantes) d'un point initial arbitraire et inconnu,  $x_0$  vers un point final fixé  $x_f$  (et non une région) en temps optimal, et ce, en évitant les régions interdites dans l'espace d'état continu. L'idée est de trouver, à partir d'un ensemble de vecteurs de commande disponibles, la séquence des vecteurs à appliquer et la durée d'activation de chacun d'eux pour optimiser le temps global de trajectoire.

Henzinger et al. (Henzinger et al., 1999) étudient le problème de contrôle pour une classe d'automates hybrides, i.e. les automates hybrides rectangulaires, en utilisant la théorie des jeux et en proposant par conséquent la notion de *jeux rectangulaires*. L'approche formelle est une généralisation de l'approche de commande basée sur les jeux temporisés décrite ci-dessus. Les auteurs montrent que, pour des jeux rectangulaires avec des objectifs exprimés en LTL (Linear temporal logic), les états gagnants pour chaque joueur (contrôleur vs. système) peuvent être calculés, et des stratégies gagnantes peuvent être synthétisées. Le problème de contrôle LTL pour les jeux hybrides se traduit comme suit : Etant donné un jeu hybride  $B$  et une formule LTL  $\varphi$  sur un état discret de  $B$ , y a-t-il une stratégie pour le joueur 1 telle que tous les résultats possibles du jeu satisfont  $\varphi$ ? Le résultat important de cette approche est que le problème de contrôle LTL peut être décidable pour les jeux hybrides.

Dans (Dang, 2000), l'auteur étudie le problème de synthèse de contrôleurs à commutation pour les systèmes hybrides. Le procédé commandé se présente comme le montre la figure 2.13. Le procédé peut être dans l'un des « modes », dans lequel son comportement est régi par des dynamiques continues distinctes. Le système peut commuter d'un mode à un autre dans

certaines zones de l'espace d'état continu  $X$ . Le choix entre les modes est fait par un contrôleur discret, qui observe de façon continue l'état du système et décide continuellement quel mode sélectionner. Par conséquent, le contrôleur peut être modélisé par un automate avec un ensemble  $Q$  d'états, où chaque état est identifié avec un mode du procédé. L'auteur propose un algorithme effectif pour la synthèse de contrôleurs à commutation pour les automates hybrides avec des dynamiques linéaires continues, sujets à des besoins de sûreté. Ceci en utilisant les techniques d'atteignabilité approximatives. Il présente aussi une adaptation de l'algorithme pour des systèmes avec des commutations incontrôlables et une méthode pour garantir que le système synthétisé soit non-Zénon.

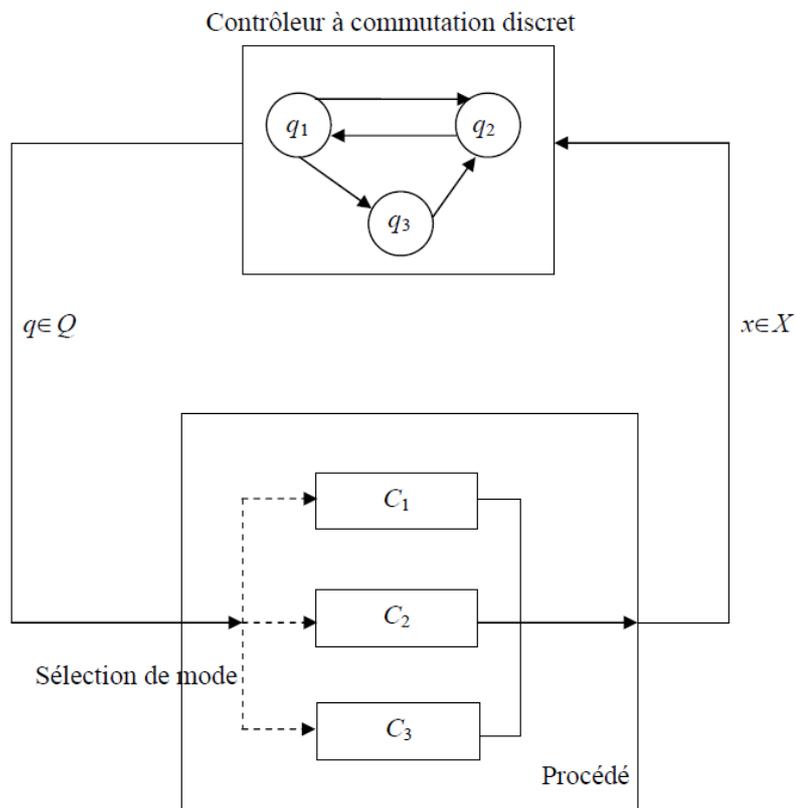


Fig.2.13- Approche de Dang.

Dans (Kurovsky, 2002), l'auteur propose une méthodologie de synthèse de la commande pour les systèmes hybrides permettant de calculer l'ensemble de toutes les lois de commande de telle sorte que le fonctionnement du système respecte les spécifications imposées par le cahier des charges. L'approche consiste à représenter la dynamique continue par un système linéaire discrétisé et la dynamique événementielle par un automate à états finis. L'ensemble donne un automate hybride sur lequel des techniques d'atteignabilité sont appliquées. On obtient l'automate atteignable qui ne contient que les trajectoires possibles du système pour

une condition initiale donnée. L'utilisation du temps discrétisé permet d'obtenir un automate à états finis modélisant le système hybride. Ce modèle est obtenu par le *dépliage temporel* de la dynamique continue du système dans chaque sommet de l'automate hybride. Les trajectoires du système hybride sont explicitement représentées par ce modèle. L'approche de synthèse de la commande est basée sur une extension de la théorie classique de la commande supervisée. Le modèle de commande synthétisé est représenté par un automate temporisé qui indique les dates d'occurrence auxquelles les événements contrôlables intervenant dans le fonctionnement du système doivent être exécutés. Ainsi, on s'affranchit de l'aspect hybride du système. Les résultats de la synthèse obtenus sont optimaux. Les différentes étapes de cette approche sont données dans la figure 2.14. Les principaux inconvénients de cette approche sont la discrétisation du temps et le problème de la terminaison de l'algorithme.

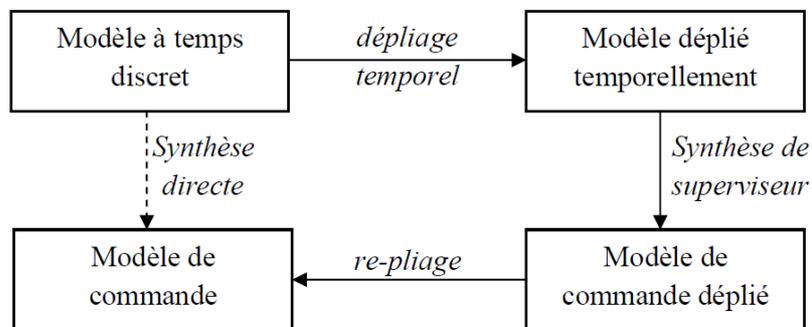


Fig.2.14- Approche de Kurovszky.

## 2.5 Conclusion

Les travaux autour de la commande des systèmes dynamiques hybrides sont récents et l'état de l'art ne permet pas aujourd'hui de présenter de manière unifiée cette problématique. Le fait que les systèmes hybrides font intervenir deux types de dynamiques, une dynamique continue et une autre discrète, implique une complexité plus importante du problème de synthèse pour cette classe de systèmes.

Dans ce chapitre, nous avons présenté quelques unes des approches de commande de ces systèmes.

Une première partie des approches de la commande des SDH porte sur la formulation et la résolution d'un problème de commande optimale caractéristique des systèmes continus. Ces approches posent le problème de commande de SDH de manière générale. Cependant, la

solution proposée n'est pas facile à exploiter et des algorithmes numériques efficaces restent à développer. Ces méthodes à forte composante continue mènent à beaucoup de résultats mais sont loin de notre travail de recherche.

La deuxième partie des approches est basée sur le principe de recherche d'une stratégie discrète qui permette de restreindre le comportement du système pour satisfaire certaines spécifications imposées. Cependant, les approches globales dans cette partie n'ont jamais été clairement établies. De plus, la plupart des travaux réalisés sont faits soit par des informaticiens qui sont loin des préoccupations des automaticiens, soit par des automaticiens mais dans ce cas, la recherche n'en est qu'à ses débuts.

Notre travail de recherche s'inscrivant dans cette dernière catégorie, nous avons jugé nécessaire d'aborder la commande des systèmes à événements discrets dans une première partie de ce chapitre. Nous avons alors présenté certaines approches de commande de ces systèmes à base de modèles logiques dans un premier temps, qui ont été ensuite étendues aux modèles temporisés.

Le problème général de synthèse de la commande des SDH peut être formulé comme suit : étant donné un système dynamique hybride, trouver une manière systématique d'intervenir dans son évolution telle que les contraintes imposées sur les variables continues soient toujours vérifiées.

Considérons à nouveau l'exemple des deux tronçons de circulation. Ce système peut être modélisé par un automate hybride rectangulaire, où on définit dans chaque état le flux des voitures et pour chaque transition, une spécification temporelle indiquant la durée des feux. On désire alors imposer des contraintes sur cet automate afin de limiter le nombre des voitures dans les tronçons à chaque changement de feux. Cet exemple sera détaillé dans les prochains chapitres.

De manière générale, notre objectif est de déterminer une commande d'une classe de systèmes dynamiques hybrides, à forte composante événementielle, modélisés par un automate hybride rectangulaire. On souhaite imposer des contraintes sur les variables d'état du système au niveau des gardes de transitions et par conséquent déterminer le comportement permettant le respect de ces spécifications. Il faudra autant que possible sélectionner la meilleure solution au sens de la maximale permissivité. Une présentation détaillée de notre approche sera faite dans les chapitres suivants.



# Chapitre 3

## Synthèse de commande maximale permissive. Calcul pour un sommet

### 3.1 Introduction

Après avoir présenté les systèmes dynamiques, leurs outils de modélisation ainsi que leurs différentes approches de commande développées dans la littérature, nous consacrons ce chapitre et le suivant au développement de notre propre démarche de commande d'une classe de systèmes dynamiques hybrides (SDH).

Notre approche repose sur la modélisation du comportement complet du système à travers une sous-classe simple d'automates hybrides : les automates hybrides rectangulaires. Ce formalisme permet d'approximer les comportements de la dynamique continue du système à travers l'utilisation de conditions de flux rectangulaires de la forme  $\dot{x} \in [a, b]$ .

Afin d'étudier le comportement dynamique du système en boucle ouverte, une analyse d'atteignabilité de l'automate construit est réalisée. Celle-ci donne tous les espaces d'état atteignables du système. Ensuite, la procédure de commande débute par l'addition des spécifications désirées par l'opérateur sur les variables du système (appelées contraintes) aux gardes de transitions de l'automate. Une fois les contraintes ajoutées et les nouveaux espaces d'état pour chaque sommet de l'automate déterminés, notre approche de calcul de la

commande se fera par une procédure amont/aval menant à de nouvelles valeurs des gardes de transition.

Dans ce domaine de recherche, les idées formelles sont souvent complexes. C'est pour cela que nous allons commencer ce chapitre par une présentation intuitive. Nous développerons ensuite notre approche d'une façon locale, c'est-à-dire autour d'un seul sommet. La démarche globale sera traitée dans le chapitre suivant.

Ce chapitre est organisé en quatre parties comme suit : dans la première partie, nous présentons un exemple intuitif pour introduire nos principales idées. La deuxième partie est consacrée à la définition formelle du modèle automate hybride rectangulaire et l'explication des notions d'analyse d'atteignabilité et d'espaces d'état atteignables. La troisième partie est dédiée à la présentation de l'automate contraint et de la notion de convexité. Enfin, dans la quatrième partie, nous décrivons notre procédure de commande pour un sommet donné de l'automate.

### 3.2 Présentation intuitive

Afin de faciliter la compréhension de notre approche de commande, nous introduisons une présentation intuitive basée sur un exemple simple.

#### 3.2.1 Système tronçon de circulation

Dans le chapitre précédent, nous avons introduit le système à deux tronçons de circulation. Cette présentation intuitive ne tiendra compte que d'un seul tronçon. Le cas de deux tronçons, plus complexe, sera étudié dans le chapitre suivant.

On considère un tronçon de circulation qui peut tolérer un nombre maximal de 150 voitures (en supposant que la distance moyenne entre deux voitures est  $L=4m$ ). Les voitures arrivent au tronçon avec une vitesse moyenne de 30 km/h et repartent avec une vitesse moyenne de 48 km/h. L'exemple est illustré dans la figure 3.1.

On suppose que le trafic est régulé soit par un feu, soit par un agent de circulation. Deux situations sont possibles : circulation autorisée ( $V_2=48$  km/h) et circulation interdite ( $V_2=0$ ).

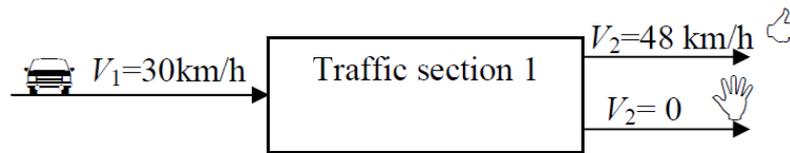


Fig.3.1- Le système tronçon de circulation.

### 3.2.2 Modélisation

Le modèle de base est un automate hybride rectangulaire composé de sommets et de transitions entre les sommets. Il sera défini formellement dans la section 3.3.

La transition d'un sommet à un autre peut être faite à n'importe quel moment déterminé par un intervalle d'occurrence d'un événement ; celui-ci sera défini par une horloge, notée  $h$  et exprimée en minutes. Initialement, le nombre de voitures est égal à 80. Alors, l'exemple peut être représenté par l'automate de la figure 3.2, en prenant comme variable d'état le nombre de voitures dans le tronçon noté  $x_1$ , sachant que le flux des voitures est donné par l'expression suivante :  $\dot{x}_1 = \frac{V_1 - V_2}{L}$ .

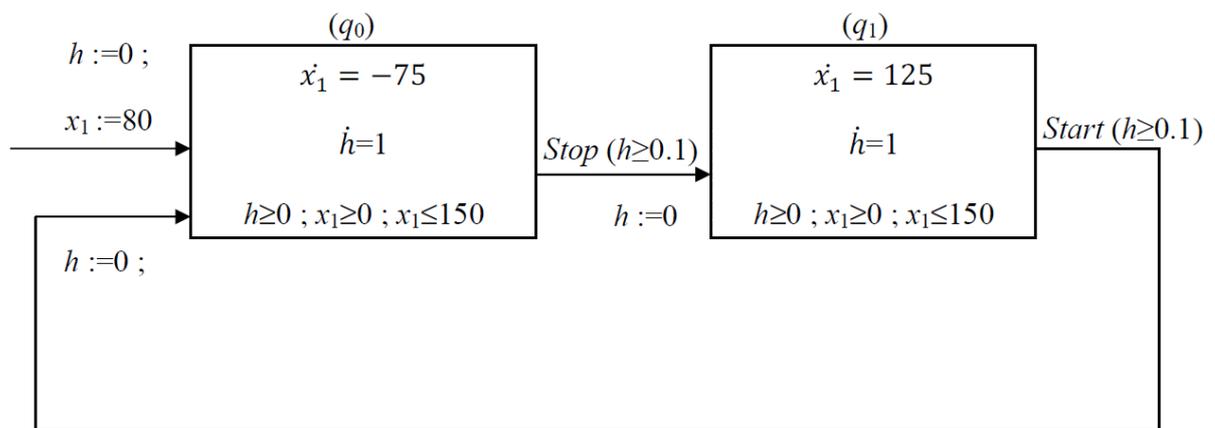


Fig.3.2- Le modèle automate du tronçon de circulation.

Chaque sommet de l'automate est défini par : les variables ( $x_1$  et  $h$ ), les dynamiques des variables (exemple:  $\dot{x}_1 = -75$ ) et les contraintes sur les variables (appelées invariants). Par exemple, l'invariant  $x_1 \leq 150$  représente la capacité maximale du tronçon.

Les événements « *Stop* » et « *Start* » sont contrôlables, c'est-à dire que leur occurrence peut être forcée dans leur intervalle d'existence.

La borne minimale 0.1 minutes relative à l'horloge  $h$  dans les gardes de transition exprime la durée minimale d'allumage des feux.

### 3.2.3 Spécifications de commande

A présent, on désire ajouter une commande dans le but de limiter le nombre de voitures dans le tronçon. L'agent de circulation ne veut pas avoir plus de 80 voitures qui attendent dans le tronçon. Dans ce cas, il autorise la circulation lorsque le nombre de voiture dans le tronçon est inférieur à 80 ( $x_1 \leq 80$ ). La spécification désirée est ajoutée à l'automate dans la transition en sortie du sommet ( $q_1$ ) comme le montre la figure 3.3. Cet automate sera appelé automate contraint.

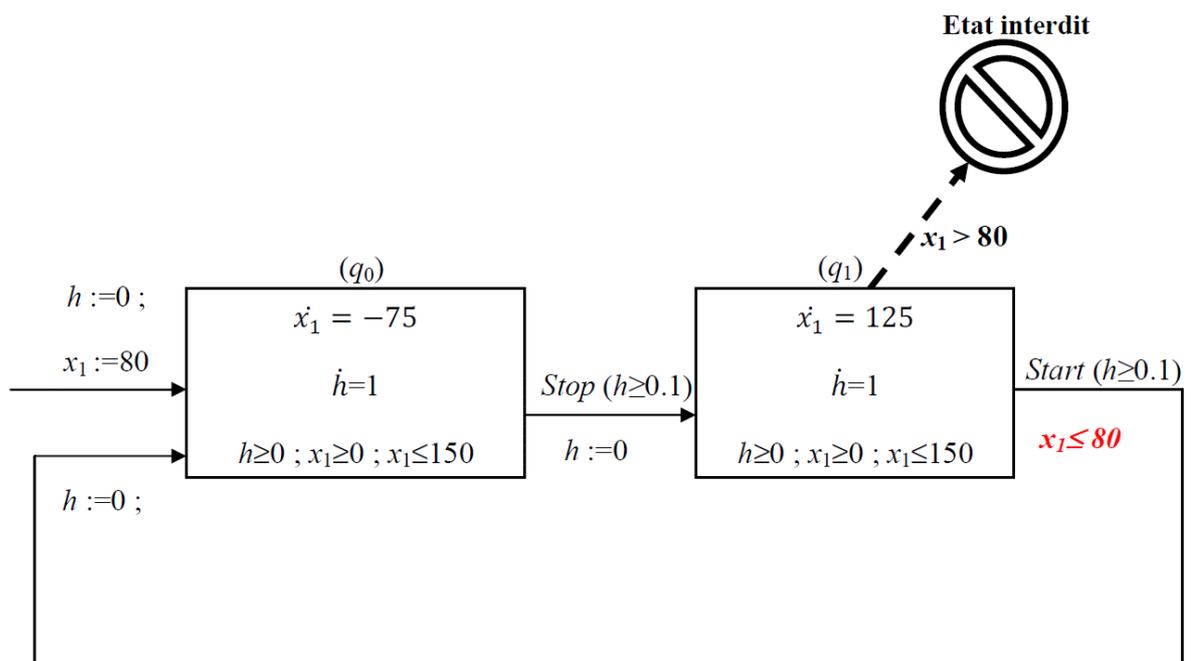


Fig.3.3- L'automate contraint du tronçon de circulation.

Dans le but de satisfaire cette spécification souhaitée, il est nécessaire de calculer les nouvelles gardes de transition ; ceci représente le problème de commande demandé.

Cependant, les spécifications sont imposées indépendamment de la dynamique du système, elles peuvent alors ne pas être atteignables. Dans ce cas, on sera amené à réduire les gardes de transition.

Le problème revient à calculer les nouvelles conditions de franchissement des transitions de façon à ce que les spécifications soient respectées tout en gardant un comportement maximal permissif. Le calcul réalisé repose sur la détermination de la durée de séjour, valeur contrainte par tout l'espace atteignable du sommet correspondant.

### 3.2.4 Contrôlabilité des variables et commande temporisée

Les variables intervenant dans un automate hybride rectangulaire (AHR) sont de deux natures; celles associées aux sommets sont continues et celles associées aux transitions peuvent être soit continues (par exemple le nombre de voitures dans le tronçon) soit discrètes (par exemple les événements « *Stop* » et « *Start* »).

Les gardes de transition dans un AHR peuvent alors portées soit sur l'état continu, soit sur un événement discret.

#### Hypothèse 3.1.

- L'état continu n'est ni contrôlable, ni observable.
- L'événement discret, qui est défini par un intervalle de temps, est à la fois contrôlable et observable.

□

Par exemple, on imagine bien que le nombre de voitures dans le tronçon n'est ni contrôlable ni observable. Par contre, l'action d'allumer un feu est à la fois contrôlable et observable.

Souvent, dans les systèmes à flux continu, on ne peut ni mesurer ni commander l'état. Par contre, on peut décider à tout instant de l'occurrence de l'événement discret dans son intervalle d'existence. Il s'en suit naturellement que l'on s'intéressera ici à une commande temporisée, d'où la modélisation explicite d'une horloge associée à chaque événement contrôlable.

Toute cette approche repose sur l'outil de modélisation qu'est l'automate hybride rectangulaire. Dans ce qui suit, nous allons le définir de manière formelle.

### 3.3 Les Automates Hybrides Rectangulaires

Dans cette partie, nous introduisons le modèle Automate Hybride Rectangulaire (AHR), qui constitue le cadre de modélisation de l'approche de commande que nous désirons développer dans ce chapitre. Ce modèle peut être considéré comme une généralisation du modèle automate temporisé, où l'évolution continue n'est pas uniquement représentée par des horloges mais par des dynamiques constantes données par des intervalles sur les variables continues du système.

Une présentation formelle du modèle automate hybride rectangulaire est proposée dans la suite de cette section, avec un rappel sur les aspects syntaxique et sémantique de ce modèle. Puis, une méthode permettant l'analyse comportementale d'un SDH modélisé par un AHR est introduite. Cette méthode repose sur l'application d'une fonction d'analyse d'atteignabilité en avant sur le modèle considéré.

#### 3.3.1 Syntaxe

Soit  $X = \{x_1, \dots, x_n\}$  un ensemble de variables réelles. On note par  $\dot{X} = \{\dot{x} | x \in X\}$  l'ensemble des dérivées premières des variables dans  $X$  par rapport au temps. Une variable  $x$  est dite *chronomètre*, si  $\dot{x} \in \{0, 1\}$ , et elle est dite horloge et sera notée  $h$ , si  $\dot{h} = 1$ .

##### Définition 3.1.

- Une *inégalité rectangulaire* sur  $X$  est une inégalité de la forme  $x \sim c$ , où  $x \in X$ ,  $c \in \mathbb{Z}$  et  $\sim \in \{<, \leq, =, \geq, >\}$ .
- Un *prédicat rectangulaire* sur  $X$  est une conjonction d'inégalités rectangulaires sur  $X$ . On note par  $Rect(X)$  l'ensemble des prédicats rectangulaires sur  $X$ .
- Une *inégalité polyédrale* sur  $X$  est une inégalité de la forme  $c_1x_1 + \dots + c_kx_k \sim c$ , où  $x_1, \dots, x_k \in X$  et  $c_1, \dots, c_k \in \mathbb{Z}$ .
- Un *prédicat polyédral* (ou un *polyèdre*) sur  $X$  est une conjonction d'inégalités polyédrales sur  $X$ . On note par  $\Psi(X)$  l'ensemble des prédicats polyédraux sur  $X$ .

- On note par  $Pred(X)$  l'ensemble des prédicats sur  $X$ , quelque soit leur origine.

**Définition 3.2.** Une valuation sur  $X$  est un vecteur de  $\mathbb{R}^n$ ,  $\mathbf{v}=(v_1, \dots, v_n)$ , qui définit une valeur  $v_i \in \mathbb{R}$  pour chaque variable  $x_i \in X$ .

**Définition 3.3.** Une *région* sur l'ensemble  $X$  est un sous ensemble de  $\mathbb{R}^n$ . Pour chaque région  $z$  et  $x_i \in X$ ,  $z(x_i) = \{v_i \in \mathbb{R} \mid \mathbf{v} \in z\}$ . On note par  $\llbracket \psi \rrbracket$  la région composée par l'ensemble des vecteurs  $\mathbf{v} \in \mathbb{R}^n$ , où le prédicat  $\psi$  est vrai si on remplace  $x_i$  par la valeur correspondante  $v_i$ , pour chaque  $i \in \{1, \dots, n\}$ .

□

En l'absence de toute ambiguïté, on peut désigner une région  $\llbracket \psi \rrbracket$  par l'expression du prédicat correspondant  $\psi$ . Pour un prédicat rectangulaire  $\psi$  et une variable  $x_i$ , on désigne par  $\llbracket \psi \rrbracket(x_i)$  l'intervalle de valeurs décrit pour les  $i^{ème}$  composants des vecteurs vérifiant  $\mathbf{v} \in \llbracket \psi \rrbracket$ .

Notre modèle AHR possède des hypothèses spécifiques; sa définition est dérivée de la définition de (Henzinger et al., 1998) :

**Définition 3.4.** Un *Automate Hybride Rectangulaire* est un octuplet  $H=(Q, X \cup \{h\}, \Sigma, T, inv, flux, q_0, E_0^{in})$ , où :

1.  $Q=\{q_1, \dots, q_k\}$  est un ensemble fini de sommets représentant les états discrets du système ;
2.  $X$  est un ensemble fini de variables réelles et  $h$  est une horloge globale. L'état continu du système est caractérisé à tout instant par un vecteur  $\mathbf{v}=(v_1, \dots, v_n)$  dans l'espace Euclidien  $\mathbb{R}^n$ , où  $v_i$  correspond à la valeur de la variable  $x_i$  ;
3.  $\Sigma$  est un ensemble d'événements contrôlables;  $\sigma \rightarrow [a, b] \in \{\mathbb{R}^+ \times (\mathbb{R}^+ \cup \infty)\}$ , on associe à chaque événement un intervalle d'horloge ;
4.  $T \subseteq Q \times \Sigma \times Pred(X) \times Pred(X) \times 2^X \times Q$  est un ensemble fini de transitions. Une transition  $(q, \sigma, g, r, R, q')$  correspond à un changement de sommets de  $q$  à  $q'$ , sur l'occurrence de l'événement  $\sigma$  et sous la condition  $\mathbf{v} \in \llbracket g \rrbracket$ , où le vecteur  $\mathbf{v}$  correspond aux valeurs courantes des variables de  $X$ . Après le franchissement de la transition,

chaque variable  $x_i \in R$  est réinitialisée, d'une manière non déterministe, pour une valeur dans l'intervalle  $\llbracket r \rrbracket(x)$ . les autres variables qui ne sont pas dans  $R$ , restent inchangées.

5.  $inv : Q \rightarrow Rect(X \cup \{h\})$  est une fonction qui associe à chaque sommet  $q \in Q$  une contrainte rectangulaire pour chaque variable  $x_i \in X$  et pour l'horloge  $h$ . Le système peut séjourner dans un sommet tant que l'invariant du sommet est satisfait ; i.e., la valeur de chaque variable  $x_i \in X$  est incluse dans l'intervalle  $\llbracket inv(q) \rrbracket(x_i)$  ;
6.  $flux : Q \rightarrow Rect(\dot{X} \cup \{\dot{h} = 1\})$  est la fonction qui affecte à chaque sommet une représentation pour l'évolution continue. Durant le séjour dans un sommet  $q$  de l'automate, l'évolution des variables continues est exprimée généralement sous la forme d'un équation d'état  $flux(q)$ , où pour chaque variable  $x_i \in X$ , sa première dérivée par rapport au temps  $\dot{x}_i$  doit évoluer dans l'intervalle  $\llbracket flux(q) \rrbracket(\dot{x}_i)$  ;
7.  $q_0$  est le sommet initial ;
8.  $E_0^{in}$  est l'espace d'entrée initial de  $q_0$  ;  $E_0^{in}$  est une région convexe (nous présenterons la notion de convexité plus tard dans ce chapitre).

□

Les hypothèses spécifiques sont :

- A chaque événement on associe un intervalle d'horloge :  $\sigma \rightarrow [a, b] \in \{\mathbb{R}^+ \times (\mathbb{R}^+ \cup \infty)\}$ .
- Tous les événements sont contrôlables. Ceci signifie qu'on est capable de modifier leur date d'occurrence dans leur intervalle d'horloge.

**Remarque 3.1.** Dans le but de simplifier la présentation de notre approche, on suppose que chaque sommet possède une seule transition de sortie. Nous verrons dans le chapitre suivant que l'approche sera facilement généralisée au cas de transitions de sortie multiples.

□

**Définition 3.5.** Un AHR admet un *non-déterminisme borné* (Henzinger et al., 1998), si :

1. les régions associées à  $E_0^{in}$  et  $flux(q)$  sont bornées pour chaque  $q \in Q$  ;
2. pour chaque transition  $(q, \sigma, g, r, R, q') \in T$  et  $x_i \in R$ , l'intervalle  $\llbracket r \rrbracket(x_i)$  est borné.

□

Dans la suite de notre travail, nous considérons uniquement les AHRs ayant un non-déterminisme borné.

Par exemple, l'automate correspondant au système tronçon de circulation donnée par la figure 3.2 est modélisé par un automate hybride rectangulaire, caractérisé par la variable continue  $x_1$  et l'horloge globale  $h$ . L'exemple correspond parfaitement à la Définition 3.4. Les événements *Start* et *Stop* sont contrôlables.

### 3.3.2 Sémantique

A chaque instant, l'état d'un automate hybride rectangulaire est donné par une paire  $(q, \nu)$  correspondant à l'association d'un état discret du système  $q$  et d'un vecteur  $\nu \in \mathbb{R}^n$  indiquant la valeur courante de chaque variable.

On présente la sémantique d'un automate hybride rectangulaire  $H$ , en termes de tous les comportements qui peuvent être générés par l'évolution du système modélisé. Dans ce sens, la sémantique d'un système continu, définie par un ensemble d'équations différentielles, est donnée par l'ensemble de toutes ses solutions, notamment de toutes ses trajectoires.

L'évolution d'un modèle automate hybride rectangulaire est réalisée selon deux types de transitions :

- Transitions continues (figure 3.4-a) : la partie discrète de l'état reste constante dans le même sommet  $q$ , tandis que la partie continue évolue de la valuation  $\nu$  à la valuation  $\nu'$ . Cette évolution se fait via une trajectoire continue, décrite par les dynamiques des variables évoluant dans des intervalles de flux spécifiés par  $flux(q)$ , tout en respectant les contraintes d'invariants décrites par  $inv(q)$ . Cette transition est notée par  $(q, \nu) \xrightarrow{\delta} (q, \nu')$ , où  $\delta \in \mathbb{R}_+$  désigne le temps écoulé durant la transition ;
- Transitions discrètes (figure 3.4-b) : elles correspondent aux changements discrets et instantanés de sommets comme décrit par le quintuplet  $t = (q, \sigma, g, r, R, q') \in T$ . Cette transition est notée par  $(q, \nu) \xrightarrow{t} (q', \nu')$ . Une transition ne peut être franchie que si le prédicat de la garde est satisfait par la partie continue de l'état :  $\nu \in \llbracket g \rrbracket$ . Dans ce cas, la partie discrète change du sommet  $q$  au sommet  $q'$  et la partie continue est mise à jour selon l'ensemble des affectations de variables.

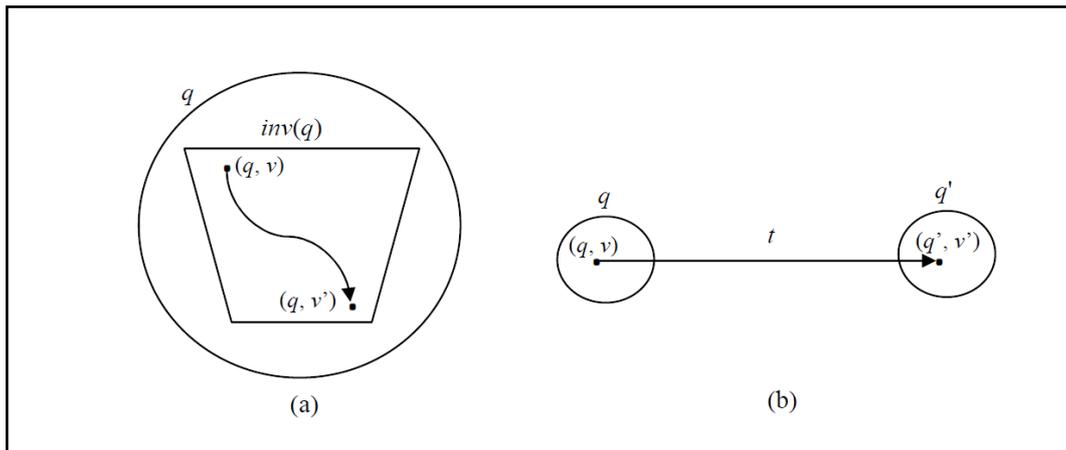


Fig.3.4- Transitions continues et transitions discrètes d'un AHR.

Ainsi, le système passe du temps dans chaque sommet discret, permettant aux variables continues d'évoluer, et effectue des transitions discrètes entre ces états. La dynamique associée aux variables continues est spécifiée par la fonction de flux. Le franchissement d'une transition discrète est synchronisé avec l'occurrence d'un événement. La valuation des variables dans  $X$  lors de ce franchissement doit vérifier la condition de garde définie.

**Définition 3.6.**

- Une *exécution* d'un AHR  $H$  est une séquence finie ou infinie de transitions continues et/ ou discrète  $(q_0, \mathbf{v}_0) \rightarrow (q_1, \mathbf{v}_1) \rightarrow (q_2, \mathbf{v}_2) \rightarrow \dots$ , où  $(q_0, \mathbf{v}_0) \in E_0^{in}$ .
- Un état  $(q, \mathbf{v})$  est dit atteignable, s'il existe une exécution de  $H$  qui rejoint cet état à partir d'un état initial  $(q_0, \mathbf{v}_0) \in E_0^{in}$ .

**3.3.3 Analyse d'atteignabilité d'un Automate Hybride Rectangulaire**

L'analyse d'atteignabilité constitue un problème central dans la vérification des propriétés des systèmes hybrides modélisés par des automates hybrides rectangulaires. Cette analyse est généralement basée sur le calcul d'un ensemble d'états accessibles à partir d'un ensemble d'états donné. En effet, on cherche à vérifier si une région de l'espace d'état est accessible à partir d'une région initiale donnée.

L'espace d'état d'un AHR est infini et non dénombrable. Par conséquent, afin de mener le calcul d'atteignabilité, il faut disposer d'un formalisme de représentation permettant de manipuler les éléments de cet espace d'état. Dans notre travail, nous avons choisi une représentation basée sur la notion de région d'état. Une *région* (ou *état symbolique*) d'un AHR est représentée par une paire  $(q, z)$ , où  $q$  correspond à un sommet d'automate et  $z$  une région de l'espace d'état continu, représenté par un polyèdre. Un état  $(q', \nu)$  est inclus dans une région  $(q, z)$  si  $q=q'$  et  $\nu \in z$ . En effet, un état symbolique  $(q, z)$  permet de représenter l'ensemble d'état  $\{(q, \nu) \mid \nu \in z\}$ .

L'analyse d'atteignabilité pose le problème de l'existence d'une trajectoire à partir d'une région initiale  $(q_0, z_0)$  qui peut amener l'état du système dans une région finale (ou marquée)  $(q_m, z_m)$ . Dans la littérature, la méthode d'analyse en avant est généralement utilisée pour vérifier l'atteignabilité d'une région de l'espace d'état. Il s'agit de calculer, d'une façon itérative, l'ensemble de tous les successeurs des états d'une région initiale  $(q_0, z_0)$ . L'ensemble des états cibles (les états marqués qu'on souhaite atteindre) est donné par  $(q_m, z_m)$ .

Les successeurs d'un état symbolique sont calculés itérativement, en alternant le calcul de successeurs continus et discrets (Roux et Rusu, 1996) (figure 3.5).

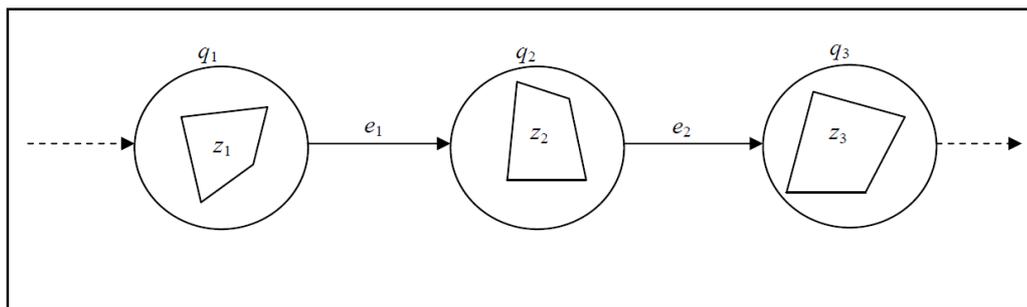


Fig.3.5- Analyse d'atteignabilité symbolique d'un AHR.

Le calcul du successeur continu d'un état symbolique  $(q, z)$  est réalisé en utilisant l'opérateur  $Post_c$  défini comme suit :

$$Post_c((q, z)) = \{(q, \nu') \mid (q, \nu) \xrightarrow{\delta} (q, \nu'), \nu \in z, \delta \in \mathbb{R}_+\}$$

De même, l'opérateur successeur discret d'une région  $(q, z)$  sur le franchissement d'une transition  $t \in T$ , noté  $Post_d((q, z), t)$ , est défini de la manière suivante :

$$Post_d((q, z), t) = \{(q', v') \mid (q, v) \xrightarrow{t} (q', v'), v \in z\}$$

On définit l'opérateur  $Post$  qui correspond à la composition de l'opérateur successeur discret, sur le franchissement d'une transition  $t \in T$ , et l'opérateur successeur continu :

$$Post((q, z), t) = Post_c((q, z)) \circ Post_d((q, z), t)$$

Pratiquement, le calcul d'une région  $(q', z')$ , successeur d'une région  $(q, z)$  sur une transition  $t = (q, \sigma, g, r, R, q')$ , est effectué en appliquant les étapes suivantes (Guéguen et Zaytoon, 2004) :

- 1- calculer  $z_1$  qui correspond à l'intersection de l'invariant du sommet  $q$  ; i.e.,  $inv(q)$ , et le futur de la région  $z$ . Le futur d'une région correspond à l'ensemble des états accessibles à partir de la région de départ en appliquant la fonction de flux (voir l'exemple de la figure 3.6) ;
- 2- calculer  $z_2$  qui correspond à l'intersection de  $z_1$  avec la condition de garde  $g$  ;
- 3- calculer  $z_3$  en appliquant la fonction de réinitialisation définie par  $R$  et  $r$  sur la région  $z_2$  ;
- 4- calculer  $z_4$  qui correspond à l'intersection de la région  $z_3$  et l'invariant du sommet  $q'$  ;  $inv(q')$  ;
- 5- enfin calculer  $z'$  qui correspond à l'intersection de l'invariant du sommet  $q'$  ;  $inv(q')$ , et le futur de la région  $z_4$ .

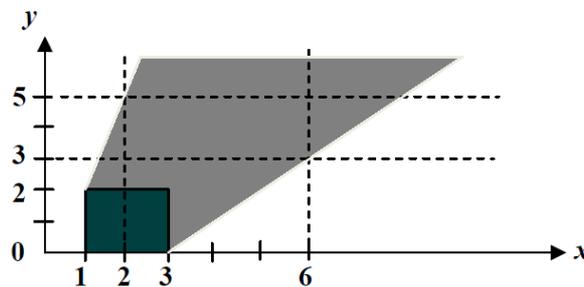


Fig.3.6- Futur d'un polyèdre  $\langle x \in [1, 3] \wedge y \in [0, 2] \rangle$  sur une fonction de flux  $\langle \dot{x} = 1 \wedge \dot{y} \in [1, 3] \rangle$ .

Les procédures d'analyse d'atteignabilité ont été implémentées dans les logiciels dédiés à la vérification des systèmes temporisés et hybrides. Parmi ces logiciels, on peut citer Hytech (Henzinger et al. 1997) et PHAver (Frehse, 2005 ; Asarin et al., 2006). Dans notre travail, nous utilisons le logiciel PHAver. Celui-ci prévoit une syntaxe particulière pour décrire textuellement un automate hybride. Il fournit des commandes pour le calcul des ensembles d'états atteignables et des relations de simulation ainsi que des commandes pour la manipulation et l'affichage de la structure de données. Son langage est aussi convivial que possible.

Tous les calculs que nous utiliserons dans notre approche de commande commencent par la détermination formelle des espaces d'état atteignables par l'automate hybride rectangulaire. Cette détermination est basée sur l'application d'une analyse en avant (réalisée en utilisant le logiciel PHAver) de l'automate du système. On obtient alors un ensemble d'inégalités donnant les relations entre les différentes variables d'état. Cette démarche est intéressante parce qu'elle nous donne une caractérisation formelle de l'espace d'état atteint que nous manipulerons pour déterminer la commande.

Les opérateurs de l'analyse en avant sont utilisés afin de calculer toutes les trajectoires possibles du système. Ceci nous mène à déterminer les espaces d'état associés aux séjours du système dans chaque sommet du modèle. Un sommet peut être atteint avec différents espaces d'état continus à son entrée, en particulier dans le cas des cycles. Par conséquent, l'espace d'état associé à un sommet est égal à la conjonction des espaces d'état continus provenant de toutes les visites possible du sommet.

**Définition 3.7. Automate atteignable :** On appellera *automate atteignable* l'automate construit après analyse d'atteignabilité en avant de l'AHR et pour lequel chaque sommet est caractérisé par un jeu d'inégalités unique.

□

L'automate atteignable est celui obtenu par le logiciel PHAver. Revenons à l'exemple du tronçon de circulation dont le modèle automate est représenté sur la figure 3.2. L'analyse en avant de cet automate donne quatre sommets: deux associés au sommet ( $q_0$ ) (qu'on notera par ( $q_{01}$ ) et ( $q_{02}$ )) et deux associés au sommet ( $q_1$ ) (qu'on notera par ( $q_{11}$ ) et ( $q_{12}$ )). L'obtention de 4 sommets à partir de deux sommets s'explique par le fait qu'on a dans le système un

comportement transitoire et un autre permanent. Ces espaces d'état sont donnés dans la figure 3.7. Ils expriment les contraintes linéaires entre la variable  $x_1$  et l'horloge globale  $h$ .

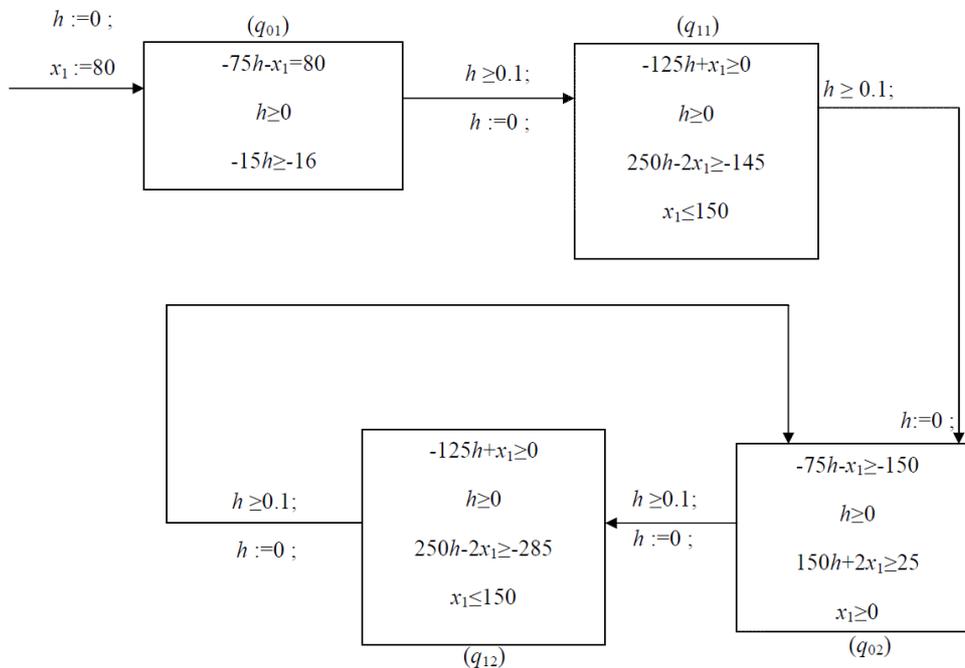


Fig.3.7- Analyse en avant de l'automate tronçon de circulation.

### 3.4 L'automate contraint

L'objectif de la commande consiste à associer des spécifications de commande au système. Cela se traduit par des contraintes sur les gardes des transitions.

Lorsque l'on impose certaines contraintes dans une transition entre deux sommets, le système commute en général selon des conditions de franchissement plus restrictives. Cela réduit l'espace d'état dans le sommet.

Ce comportement rejoint la notion d'états interdits, introduite par Wonham (Wonham, 2011) dans la théorie classique de commande par supervision.

Prenons le cas de l'automate contraint de la figure 3.3 relatif au système tronçon de circulation.

Sachant que le flux de la variable  $x_1$  est positif dans le sommet ( $q_1$ ), et puisque la contrainte imposée est telle que  $x_1 \leq 80$  à l'aval de ce sommet, alors si  $x_1 > 80$ , le système va dans l'état interdit (figure 3.8).

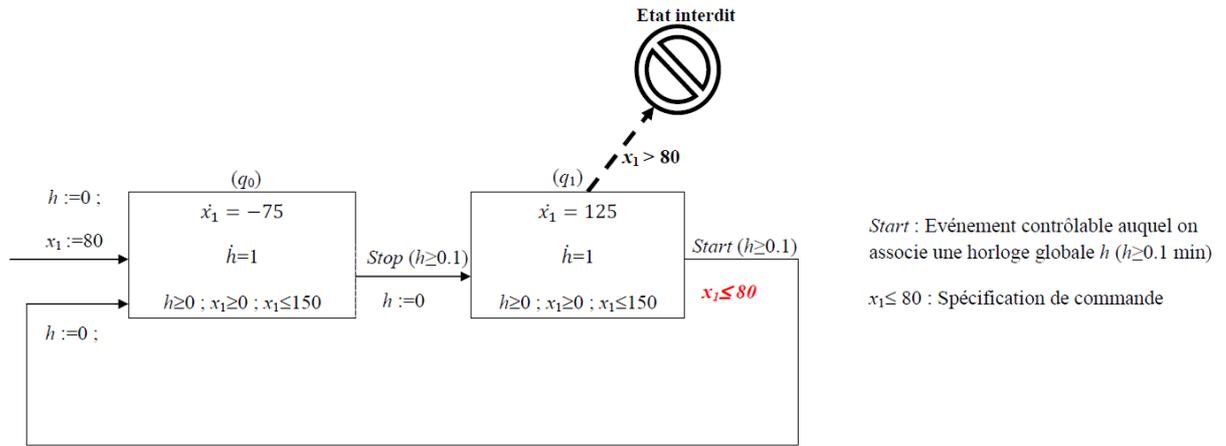


Fig.3.8- Spécification de commande.

### 3.4.1 Définitions

Dans la suite, nous allons définir les notions relatives aux spécifications de commande qui aboutiront à l'automate hybride rectangulaire contraint.

**Définition 3.8. Contrainte:** Une contrainte est une conjonction d'inégalités rectangulaires (ou polyédrales) sur  $X$  de la forme  $x \sim c$ , où  $x \in X$ ,  $c \in \mathbb{Z}$  et  $\sim \in \{<, \leq, =, \geq, >\}$  (ou  $c_1x_1 + \dots + c_kx_k \sim c$ , où  $x_1, \dots, x_k \in X$  et  $c_1, \dots, c_k \in \mathbb{Z}$ ). Cette conjonction est appelée prédicat rectangulaire (ou polyédral) sur  $X$  et elle est notée  $C(X)$ .

**Définition 3.9. Spécification de commande:** Une spécification de commande est une fonction qui associe à chaque transition une contrainte sur  $X$  :  $T \rightarrow (X \rightarrow C(X))$ .

**Définition 3.10. Automate Hybride Rectangulaire Contraint:** Un Automate Hybride Rectangulaire Contraint (AHR Contraint) est défini par l'octuplet  $H=(Q, X \cup \{h\}, \Sigma, T, inv, flux, q_0, E_0^{in})$ , comme dans la Définition 3.4, sauf pour le quatrième point qui devient :

3.  $T \subseteq Q \times \Sigma \times Pred(X) \times C(X) \times Pred(X) \times 2^X \times Q$  est un ensemble fini de transitions, où  $Pred(X)$  est un ensemble de prédicats sur  $X$  et  $C(X)$  est un ensemble de contraintes sur  $X$ .

□

On notera par  $U_i$  (sachant que l'on a supposé que chaque sommet possédait une seule transition de sortie (Remarque 3.1)) la garde de sortie du sommet  $(q_i)$ . Celle-ci contient, en

plus de la garde de l'automate non contraint, l'ensemble des contraintes sur  $X$  ajoutées à la transition correspondante.

Par exemple, dans le cas de la transition de sortie du sommet ( $q_1$ ) dans l'automate contraint du système tronçon de circulation (figure 3.3),  $U_1 = \{h \geq 0.1 ; x_1 \leq 80\}$ .

### 3.4.2 Problème d'espace convexe

La notion de convexité est un problème important car elle garantit que toutes les valeurs d'une garde de transition sont atteignables.

On considère la portion de l'automate donnée par la figure 3.9.

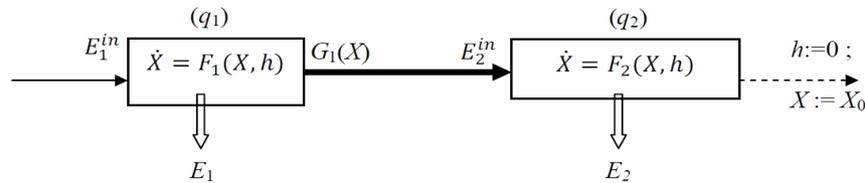


Fig.3.9- Portion d'un automate.

$G_1(X)$  est la garde sur les variables  $X$ , contenant les contraintes.

Puisque nous avons affaire à un AHR,  $F_1$  et  $F_2$  sont des applications linéaires (flux de l'AHR).

On considère que  $(E_1^{in}$  et  $E_2^{in})$  et  $(E_1$  et  $E_2)$  sont respectivement les espaces d'entrée et les espaces d'état atteignables des sommets  $(q_1)$  et  $(q_2)$ .

#### Propriété 3.1. Convexité pour un sommet

Si  $E_1^{in}$  est convexe,  $E_1$  et  $E_2^{in}$  sont convexes.

**Preuve :**

- L'image d'un ensemble convexe par une application affine est convexe (Gilbert, 2007) :  $E_1^{in}$  étant convexe  $\Rightarrow F(E_1^{in}) \Rightarrow E_1$  est convexe.
- L'intersection préserve la convexité (Gilbert, 2007) :  $E_1$  et  $G_1(X)$  étant convexes  $\Rightarrow (E_1 \wedge G_1(X)) \Rightarrow E_2^{in}$  est convexe.

□

### Propriété 3.2. Convexité et Projection

Si  $E_i$  est convexe, alors sa projection sur une variable  $x_j$  notée  $E_i^j$  est aussi convexe.

#### Preuve :

Evidente (Gilbert, 2007).

□

**Remarque 3.2.** On note par  $W_i$  l'ensemble convexe d'inégalités pour un sommet  $(q_i)$  sur les variables d'état  $x_j$  et l'horloge globale  $h$ .

$W_j^i = [v_j^i, w_j^i]$  : l'intervalle d'existence correspondant à chaque variable  $x_j$  dans le sommet  $(q_i)$ .

Les limites  $v_j^i$  et  $w_j^i$  peuvent être obtenues par programmation linéaire.

Dans ce cas, comme  $W_i$  est convexe,  $W_j^i$  est aussi convexe (Propriété 3.2).

□

### 3.4.3 Analyse d'atteignabilité de l'Automate Hybride Rectangulaire Contraint

L'étape qui précède le calcul de la commande consiste à réaliser une analyse d'atteignabilité en avant de l'automate hybride rectangulaire contraint. Cette analyse mène à une réduction de l'espace d'état, par rapport à celui obtenu sur l'automate non contraint, et elle peut aussi mener à une suppression d'état. Dans ce dernier cas, cela signifie que certains états peuvent ne pas être atteignables par les contraintes ajoutées. En effet, les contraintes sont imposées indépendamment du système. S'il y a suppression d'état, cela peut être du au fait que les contraintes soient plus fortes que la dynamique du système. Si, par contre, il n'y a pas de suppression d'état, cela signifie qu'il existe une possible commande non vide pour l'automate contraint. L'automate obtenu après analyse en avant est appelé *automate contraint atteignable*.

**Remarque 3.3.** On verra plus tard, que même s'il n'y a pas eu suppression d'états par l'ajout des contraintes, on peut être amené à réduire les spécifications de commande pour cause d'inconsistance avec les dynamiques du système.

□

Considérons l'automate contraint du système tronçon de circulation illustré par la figure 3.3. L'analyse d'atteignabilité en avant de cet automate donne l'automate contraint atteignable de la figure 3.10. Celui-ci est caractérisé par les espace d'état atteignables de l'automate contraint qui expriment les contraintes linéaires entre la variable d'état  $x_1$  et l'horloge global  $h$ . Ces contraintes fournissent une caractérisation bijective de l'espace d'état et cette formalisation va nous permettre de calculer les valeurs de l'horloge de commande.

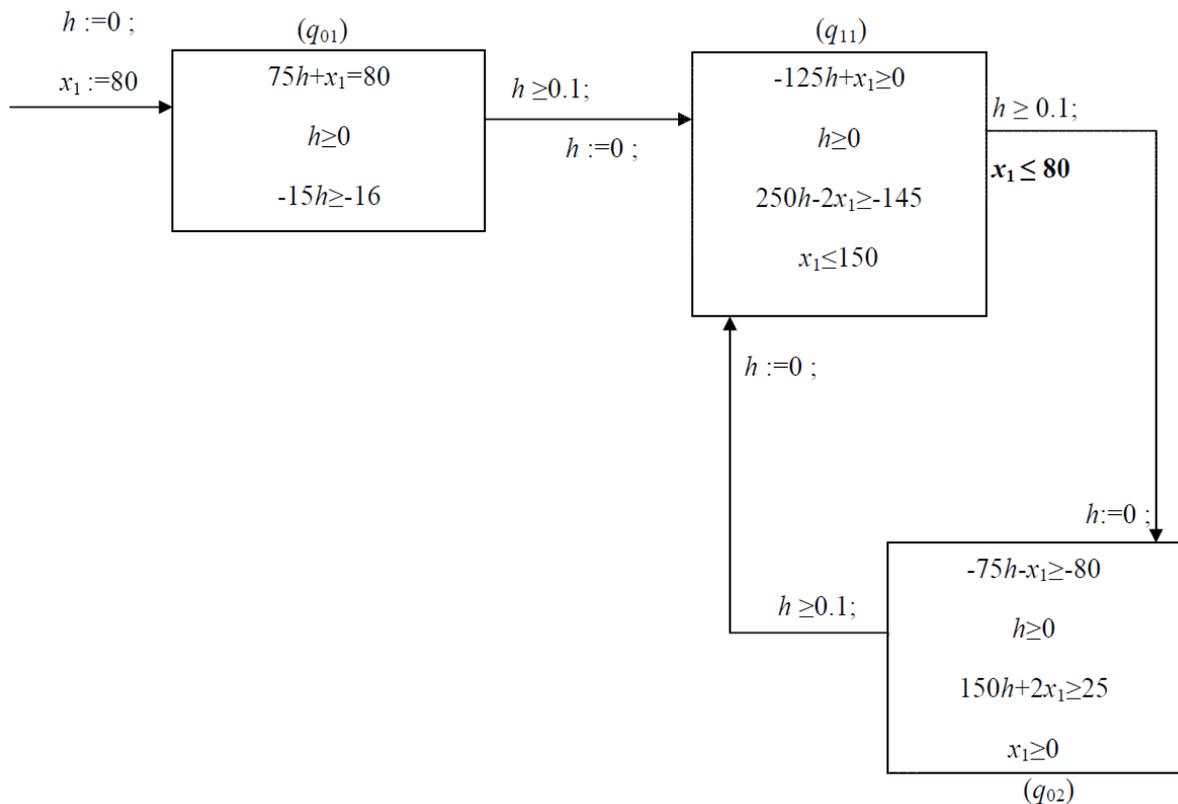


Fig.3.10- Analyse en avant de l'automate contraint.

Dans ce cas précis, nous avons obtenu trois espaces d'état, alors que l'analyse en avant de l'automate non contraint avait donné quatre espaces. Ceci est du au fait que les espaces  $(q_{11})$  et  $(q_{12})$  sont superposés, ils sont identiques.

Comparons, par exemple les espaces  $(q_{02})$  dans les deux cas étudiés. Les espaces tracés sont donnés par la figure 3.11. On constate bien que l'espace d'état a été réduit par l'ajout de la contrainte amont ( $x_1 \leq 80$ ). Cependant l'espace n'a pas été supprimé, ce qui signifie qu'il existe une possible commande permettant le respect de la contrainte imposée.

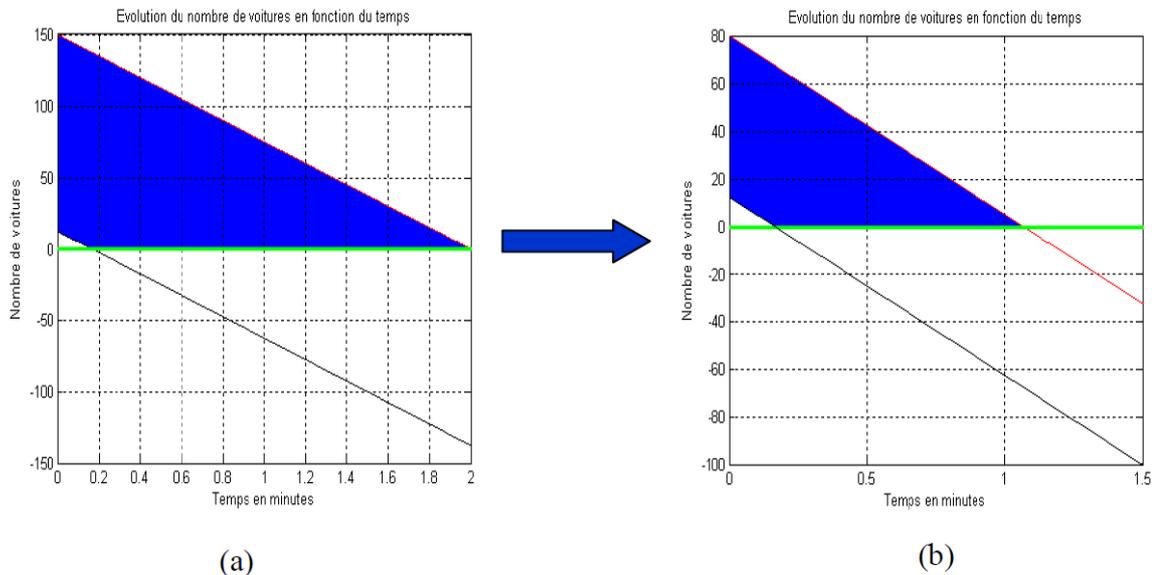


Fig.3.11- (a) Espace  $(q_{02})$  de l'automate non contraint – (b) Espace  $(q_{02})$  de l'automate contraint.

### 3.5 Calcul local d'une commande maximale permissive

Dans la suite, nous allons introduire notre approche de commande, appliquée à l'automate contraint défini ci-dessus. Notre démarche sera faite en deux étapes : une première étape focalisée sur une transition entre deux sommets, et une deuxième étape généralisée à l'automate entier. Cette dernière sera développée dans le chapitre suivant.

Trouver le superviseur maximal permissif consiste à déterminer toutes les gardes maximales de commutation qui respectent les invariants et les contraintes.

**Remarque 3.4.** On appellera *garde initiale*, la garde  $U_i$  avant le calcul de la commande, et on notera  $V_i$  la *nouvelle garde* obtenue après le calcul de la commande.

□

Les contraintes ajoutées à l'automate sont définies indifféremment du processus. Il n'y a alors aucune garantie de leur atteignabilité. En effet, elles peuvent être incompatibles avec la dynamique du sommet correspondant. Ceci revient à calculer la *durée de séjour la plus contraignante* dans le sommet en question, notée  $\delta_j$  pour chaque variable  $x_j$ . Cette valeur doit être supérieure à la *durée de séjour minimale possible*, qui correspond à la différence entre la valeur minimale de sortie de l'horloge et la valeur maximale d'entrée de l'horloge. Ce sera l'idée clé de notre approche de synthèse de la commande.

La base du calcul revient à résoudre le problème pour une transition donnée entre deux sommets  $(q_{i-1})$  et  $(q_i)$ . On suppose que la nouvelle garde  $V_i$  à l'aval du sommet  $(q_i)$  est connue et on remonte pour calculer la nouvelle garde à l'aval de  $(q_{i-1})$ .

La nouvelle garde calculée pour la transition entre  $(q_{i-1})$  et  $(q_i)$  est notée  $V_{i-1}$ . Elle est calculée à partir de la garde initiale  $U_{i-1}$  (Définition 3.9), l'espace d'état atteignable  $E_{i-1}$  et la garde aval nouvellement calculée  $V_i$ . Les principales idées sont illustrées dans la figure 3.12.

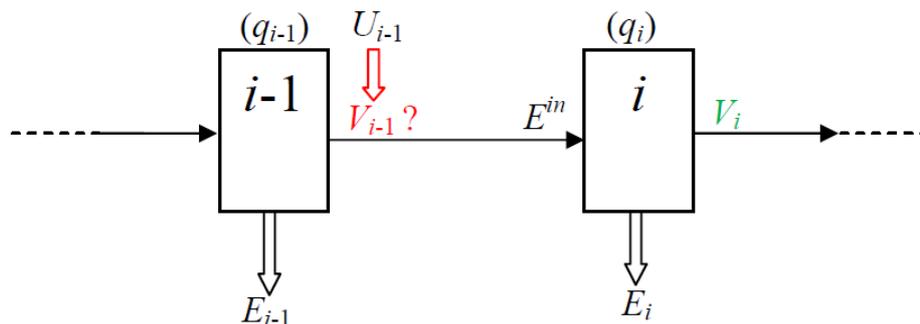


Fig.3.12- Calcul de la nouvelle garde  $V_{i-1}$ .

Avant de développer le théorème de calcul de la commande, on rappelle les principales variables dont on a besoin. Elles sont données dans la figure 3.13.

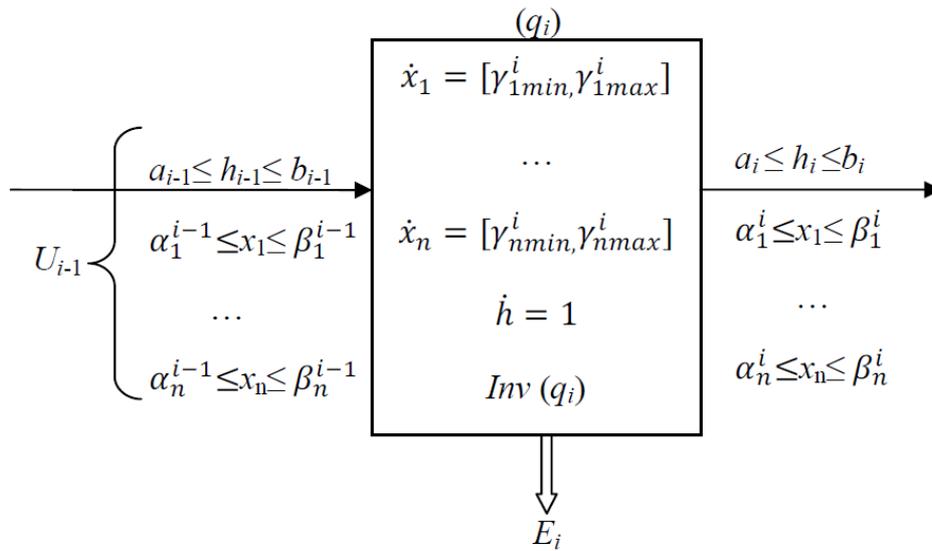


Fig.3.13- Principales variables nécessaires pour le sommet  $(q_2)$ .

Où :

- $x_j \in [\alpha_j^i, \beta_j^i]$  est l'intervalle d'existence convexe correspondant à chaque variable  $x_j$  dans le sommet  $(q_i)$ . (Remarque 3.2) ;
- $\gamma_j^i \in [\gamma_{jmin}^i, \gamma_{jmax}^i]$  est l'intervalle de flux pour la variables  $x_j$  dans le sommet  $(q_i)$ . Il est évident que les bornes du flux sont de même signe. Cette caractérisation correspond au comportement de la plupart des systèmes réels ;
- $h_{i-1} \in [a_{i-1}, b_{i-1}]$  est l'intervalle d'entrée de l'horloge au sommet  $(q_i)$  ;
- $h_i \in [a_i, b_i]$  est l'intervalle de sortie de l'horloge du sommet  $(q_i)$  ;
- $E_i$  est l'espace d'état atteignable du sommet  $(q_i)$  ;
- $U_{i-1}$  est la garde initiale à l'entrée du sommet  $(q_i)$ .

### 3.5.1 Cas d'une seule variable d'état

Pour simplifier la démonstration, on suppose dans un premier temps que le système possède une seule variable d'état, notée  $x_1$ , en plus de l'horloge globale  $h$ , comme c'est le cas pour notre système tronçon de circulation. Dans ce cas, le résultat de calcul de la commande pour la transition à l'aval du sommet  $(q_{i-1})$  est donné par le théorème 3.1.

**Théorème 3.1. Calcul de la commande pour un système à une seule variable d'état**

Soit  $V_i$  la nouvelle garde calculée à l'aval du sommet ( $q_{i(i \in [0, N])}$ ), alors la nouvelle garde  $V_{i-1}$  à l'aval du sommet ( $q_{i-1}$ ), pour un système à une seule variable d'état  $x_1$ , est calculée comme suit :

- Cas où  $\gamma_1^i > 0$ :  $V_{i-1} = E_{i-1} \wedge U_{i-1} \wedge [0, \beta_1^i - \gamma_{1max}^i(a_i - b_{i-1})]$  (1)

- Cas où  $\gamma_1^i < 0$ :  $V_{i-1} = E_{i-1} \wedge U_{i-1} \wedge [\alpha_1^i + \gamma_{1min}^i(a_i - b_{i-1}), \infty]$  (2)

□

**Preuve :**

On suppose ici que  $V_i$  est connue et on veut calculer  $V_{i-1}$ .

L'espace d'entrée au sommet ( $q_i$ ) est noté  $E^{in}$ , son expression est :  $E^{in} = U_{i-1} \wedge E_{i-1}$ , tel que  $U_{i-1}$  correspond à la garde initiale associée à la transition ( $q_{i-1}$ ) → ( $q_i$ ) (figure 3.12).

On note par  $x_1^{in}$  un élément de  $E^{in}$ , et on calcule la durée de séjour la plus contraignante  $\delta_1$  dans le sommet ( $q_i$ ) pour la variable  $x_1$  :

L'expression générale de la durée de séjour dans le sommet ( $q_i$ ) pour la variable  $x_1$ , notée  $d_1$  est :

$$d_1 = \frac{x_1 - x_1^{in}}{\gamma_1^i}$$

où  $x_1 \in V_i$ .

- On étudie dans un premier temps le cas où le flux de la variable  $x_1$  dans le sommet ( $q_i$ ) est positif :  $\gamma_1^i > 0$ . Dans ce cas, la durée de séjour la plus contraignante s'écrit comme suit :

$$\delta_1 = \frac{x_{1max} - x_{1max}^{in}}{\gamma_{1max}^i}$$

Où :

- $x_{1max}$  est la valeur maximale à la sortie du sommet ( $q_i$ ) (puisque  $\gamma_1^i > 0$ ).

- $x_{1max}^{in}$  est la valeur maximale de  $x_1^{in}$ . Elle est choisie car il y a une incertitude à l'entrée du sommet et par conséquent, il faut prendre la valeur la plus pénalisante.
- $\gamma_{1max}^i$  est la valeur maximale du flux dans le sommet ( $q_i$ ). Elle est choisie car c'est celle qui donne le temps de séjour le plus petit.

Cette durée de séjour la plus contraignante doit toujours être supérieure à la durée de séjour minimale possible, qui correspond à la différence entre la valeur minimale de sortie de l'horloge et la valeur maximale d'entrée de l'horloge :  $a_i - b_{i-1}$ .

$$\Leftrightarrow \delta_1 \geq a_i - b_{i-1} \Leftrightarrow x_{1max}^{in} \leq x_{1max} - \gamma_{1max}^i (a_i - b_{i-1})$$

Reprenons la Remarque 3.2 et considérons que  $W_i = V_i$ .

Dans ce cas, on note l'intervalle convexe correspondant à la variable  $x_1$  par  $[\alpha_1^i, \beta_1^i]$ , où :

- $\alpha_1^i$  est la valeur minimale prise par  $x_1$  respectant  $V_i$ ;
- $\beta_1^i$  est la valeur maximale prise par  $x_1$  respectant  $V_i$

$x_1 \in V_i \Leftrightarrow x_1 \in [\alpha_1^i, \beta_1^i]$ . Comme  $x_{1max}^{in} \leq x_{1max} - \gamma_{1max}^i (a_i - b_{i-1})$ , alors  $x_1^{in} \in [0, \xi_1^i]$  où  $\xi_1^i \in [0, \beta_1^i - \gamma_{1max}^i (a_i - b_{i-1})]$ . Ceci correspond à un possible fonctionnement de l'automate. **Ce fonctionnement est maximal permissif lorsque  $\xi_1^i = \beta_1^i - \gamma_{1max}^i (a_i - b_{i-1})$** . D'où l'expression (1) ci-dessus.

➤ Le cas où  $\gamma_1^i < 0$  est symétrique.

$$\left. \begin{array}{l} \text{Dans ce cas : } \delta_1 = \frac{x_{1min} - x_{1min}^{in}}{\gamma_{1min}^i} \\ \delta_1 \geq a_i - b_{i-1} \end{array} \right\} x_{1min}^{in} \geq x_{1min} + \gamma_{1min}^i (a_i - b_{i-1})$$

$\Leftrightarrow x_1^{in} \in [\xi_1^i, \infty]$  où  $\xi_1^i \in [\alpha_1^i + \gamma_{1min}^i (a_i - b_{i-1}), \infty]$ . Ceci correspond à un possible fonctionnement de l'automate. **Ce fonctionnement est maximal permissif lorsque  $\xi_1^i = \alpha_1^i + \gamma_{1min}^i (a_i - b_{i-1})$** . D'où l'expression (2) ci-dessus.

□

### 3.5.2 Application au système tronçon de circulation

Il est intéressant à ce stade d'appliquer notre approche locale sur l'une des transitions de l'automate contraint du système tronçon de circulation, sachant que le système possède une seule variable d'état  $x_1$  correspondant au nombre de voitures. On choisit, par exemple, de calculer la nouvelle garde  $V_{01}$  entre les états  $(q_{01})$  et  $(q_{11})$ . Pour cela, on a besoin des informations suivantes :

- 1- L'espace d'état  $E_{01}$  du sommet  $(q_{01})$
- 2- La garde initiale  $U_{01}$  entre les états  $(q_{01})$  et  $(q_{11})$
- 3- Le flux de la variable  $x_1$  dans le sommet  $(q_{11})$
- 4- La garde en aval du sommet  $(q_{11})$

L'espace d'état  $E_{01}$  du sommet  $(q_{01})$  est celui obtenu après analyse en avant de l'automate contraint du système. Il est donné par la figure 3.10.  $E_{01}=\{75h+x_1=80 ; h \geq 0 ; 15h \leq 16\}$ .

La garde initiale  $U_{01}$  entre les états  $(q_{01})$  et  $(q_{11})$  est donnée dans l'automate contraint de la figure 3.3.  $U_{01}=\{h \geq 0.1\}$ .

Le flux de la variable  $x_1$  dans le sommet  $(q_{11})$  est positif. Il est égale à  $\dot{x}_1 = \gamma_1^{11} = 125$ .

La garde en aval du sommet  $(q_{11})$  est égale à la garde initiale  $U_{11}$ , puisque la nouvelle garde  $V_{11}$  n'a pas encore été calculée.  $U_{11}=\{h \geq 0.1 ; x_1 \leq 80\}$ .

Sachant que le flux dans  $(q_{11})$  est positif, les valeurs dont on a besoin pour le calcul de la nouvelle garde  $V_{01}$  sont :  $\beta_1^{11}=80$ ,  $a_{11}=0.1$  (valeur minimale de l'horloge à la sortie de  $(q_{11})$ ) et  $b_{01}=0$  (valeur de l'horloge à l'entrée de  $(q_{11})$ ). Elle est nulle car elle a été réinitialisée).

Calculons alors la valeur maximale permissive de  $x_1$  dans la transition entre les états  $(q_{01})$  et  $(q_{11})$ , notée  $\xi_1^{11}$ .

$$\xi_1^{11} = \beta_1^{11} - \gamma_1^{11}(a_{11} - b_{01})$$

$$\xi_1^{11} = 80 - 125(0.1 - 0)$$

$$\xi_1^{11} = 67.5$$

Finalement, on obtient la nouvelle garde maximale permissive :

$$V_{01} = E_{01} \wedge U_{01} \wedge [0, \beta_1^{11} - \gamma_1^{11}(a_{11} - b_{01})]$$

$$V_{01} = \{75h + x_1 = 80 ; h \geq 0 ; 15h \leq 16\} \wedge \{h \geq 0.1\} \wedge \{0 \leq x_1 \leq 67.5\}$$

**Remarque 3.5.** Le calcul de la nouvelle garde maximale permissive  $V_{01}$  a mené à une réduction de l'intervalle de la variable  $x_1$  à  $x_1 \leq 67.5$ . La valeur initiale est calculée intuitivement; en effet, sachant qu'initialement  $x_1 = 80$ , que le flux dans ( $q_{01}$ ) est égal à  $-75$  et que le système peut quitter le sommet au plus tôt pour  $h = 0.1$ , alors l'intervalle initial est  $x_1 \leq 72.5$ . L'idée générale de commande maximale permissive pour un sommet est résumée dans la figure 3.14.

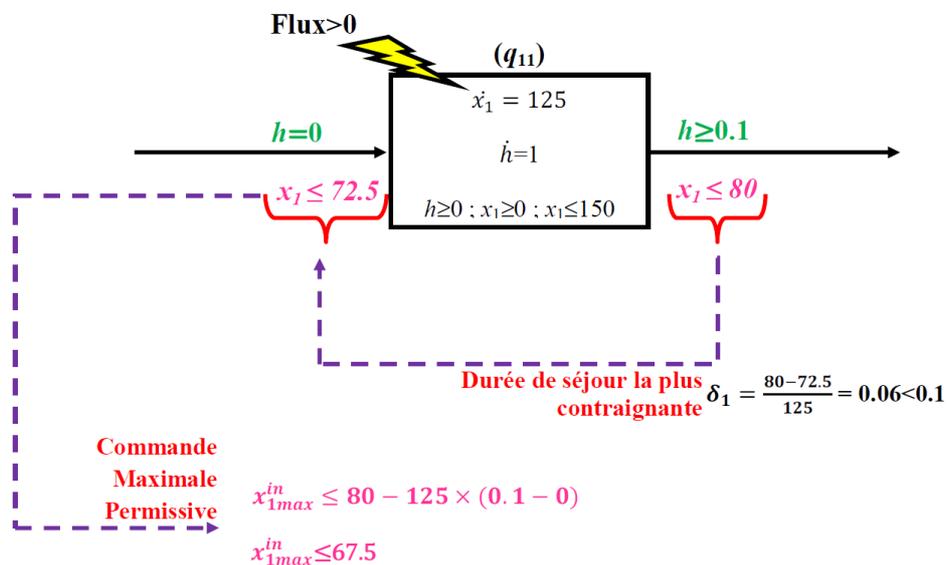


Fig.3.14- Idée générale de commande maximale permissive.

□

### 3.5.3 Généralisation au cas de plusieurs variables d'état

Dans la partie précédente, nous avons choisi de présenter notre approche locale de commande maximale permissive de la manière la plus simplifiée, c'est-à-dire celle où l'on a une seule variable d'état. On généralise à présent notre calcul au cas plus complexe de plusieurs variables d'état. Dans ce cas, nous allons nous baser sur le sommet de la figure 3.13 présenté ci-dessus et contenant  $n$  variables d'état. Notre approche de commande reste le calcul de la nouvelle garde entre deux sommets ( $q_{i-1}$ ) et ( $q_i$ ) comme indiqué dans la figure 3.12. Le

résultat de calcul de la commande pour la transition à l'aval du sommet ( $q_{i-1}$ ) est donné par le théorème 3.2.

**Théorème 3.2. Calcul de la commande pour un système à plusieurs variables d'état**

Soit  $V_i$  la nouvelle garde calculée à l'aval du sommet ( $q_{i(i \in [0, N])}$ ), alors la nouvelle garde  $V_{i-1}$  à l'aval du sommet ( $q_{i-1}$ ), pour un système à  $n$  variables d'état  $x_{j(j \in [1, n])}$ , est calculée comme suit :

- Cas où  $\gamma_j^i > 0$ :  $V_{i-1} = E_{i-1} \wedge U_{i-1} \wedge (\bigwedge_{j=1}^n [0, \beta_j^i - \gamma_{jmax}^i (a_i - b_{i-1})])$  (3)

- Cas où  $\gamma_j^i < 0$ :  $V_{i-1} = E_{i-1} \wedge U_{i-1} \wedge (\bigwedge_{j=1}^n [\alpha_j^i + \gamma_{jmin}^i (a_i - b_{i-1}), \infty])$  (4)

□

**Preuve :**

La preuve du théorème 3.2 est équivalente à celle du théorème 3.1, sauf que l'on remplace  $x_1^{in}$  la variable d'état unique par  $x_j^{in}$  ( $j \in [1, n]$ ) un élément quelconque de  $E^{in}$ . Dans ce cas, on calcule la durée de séjour la plus contraignante  $\delta_j$  dans le sommet ( $q_i$ ) pour la variable  $x_{j(j \in [1, n])}$  :

L'expression générale de la durée de séjour dans le sommet ( $q_i$ ) pour la variable  $x_j$ , notée  $d_j$  est :

$$d_j = \frac{x_j - x_j^{in}}{\gamma_j^i}$$

où  $x_j \in V_i$  et  $x_j$  la valeur de la variable à l'état de commutation.

➤ On étudie dans un premier temps le cas où le flux de la variable  $x_j$  dans le sommet ( $q_i$ ) est positif :  $\gamma_j^i > 0$ . Dans ce cas, la durée de séjour la plus contraignante s'écrit comme suit :

$$\delta_j = \frac{x_{jmax} - x_{jmax}^{in}}{\gamma_{jmax}^i}$$

Par comparaison à la durée de séjour minimale possible:

$$\delta_j \geq a_i - b_{i-1} \Leftrightarrow x_{jmax}^{in} \leq x_{jmax} - \gamma_{jmax}^i(a_i - b_{i-1})$$

En considérant  $W_i=V_i$ , on note l'intervalle convexe correspondant à la variable  $x_j$  par  $[\alpha_j^i, \beta_j^i]$ , où :

- $\alpha_j^i$  est la valeur minimale prise par  $x_j$  respectant  $V_i$  ;
- $\beta_j^i$  est la valeur maximale prise par  $x_j$  respectant  $V_i$

$x_j \in V_i \Leftrightarrow x_j \in [\alpha_j^i, \beta_j^i]$ . Comme  $x_{jmax}^{in} \leq x_{jmax} - \gamma_{jmax}^i(a_i - b_{i-1})$ , alors  $x_j^{in} \in [0, \xi_j^i]$  où  $\xi_j^i \in [0, \beta_j^i - \gamma_{jmax}^i(a_i - b_{i-1})]$ . Ceci correspond à un possible fonctionnement de l'automate. **Ce fonctionnement est maximal permissif lorsque  $\xi_j^i = \beta_j^i - \gamma_{jmax}^i(a_i - b_{i-1})$** . D'où l'expression (3) ci-dessus.

➤ Le cas où  $\gamma_j^i < 0$  est symétrique.

$$\text{Dans ce cas : } \left. \begin{array}{l} \delta_j = \frac{x_{jmin} - x_{jmin}^{in}}{\gamma_{jmin}^i} \\ \delta_j \geq a_i - b_{i-1} \end{array} \right\} x_{jmin}^{in} \geq x_{jmin} + \gamma_{jmin}^i(a_i - b_{i-1})$$

$\Leftrightarrow x_j^{in} \in [\xi_j^i, \infty]$  où  $\xi_j^i \in [\alpha_j^i + \gamma_{jmin}^i(a_i - b_{i-1}), \infty]$ . Ceci correspond à un possible fonctionnement de l'automate. **Ce fonctionnement est maximal permissif lorsque  $\xi_j^i = \alpha_j^i + \gamma_{jmin}^i(a_i - b_{i-1})$** . D'où l'expression (4) ci-dessus.

□

Dans cette formule générale, le terme ajouté est «  $\bigwedge_{j=1}^n$  » pour signifier qu'il s'agit de l'ensemble des variables d'état.

**Remarque 3.6.** L'expression générale de la commande  $V_{i-1} = E_{i-1} \wedge U_{i-1} \wedge (\bigwedge_{j=1}^n [0, \beta_j^i - \gamma_{jmax}^i(a_i - b_{i-1})])$  est constituée de deux parties :

- Une partie due à la commande avale ( $E_{i-1} \wedge U_{i-1}$ ) signifiant que l'on doit quitter l'état aussitôt que la contrainte dans  $U_{i-1}$  est vérifiée (pour ne pas aller dans l'état interdit).

- Une partie due à la commande amont ( $\bigwedge_{j=1}^n [0, \beta_j^i - \gamma_{jmax}^i(a_i - b_{i-1})]$ ) qui correspond à la satisfaction de la nouvelle garde  $V_i$ .

□

### 3.6 Conclusion

Dans ce chapitre, nous avons présenté notre approche de commande des systèmes hybrides par automates hybrides rectangulaires pour un sommet donné. Cette démarche nécessite la conception d'un modèle complet du système puis l'application d'une procédure hors-ligne pour la détermination d'une commande maximale permissive.

Les différentes étapes de notre approche de commande peuvent se résumer comme suit :

- 1- Elaboration du modèle automate hybride rectangulaire du système à commander. On y définit les variables du système, leurs évolutions rectangulaires (dans un intervalle donné) ainsi que les contraintes physiques qui délimitent leurs séjours dans un état continu. D'autre part, on caractérise aussi les transitions discrètes d'un état à un autre suite à l'occurrence d'un événement contrôlable, auquel on associe un intervalle d'horloge.
- 2- Réalisation d'une analyse d'atteignabilité de l'automate défini dans le but de déterminer les espaces d'état atteignables du système et d'étudier son comportement dynamique.
- 3- Ajout de spécifications de contraintes désirées à l'automate de départ. Ces contraintes sont imposées aux variables d'état sur les gardes de transition de l'automate. Elles sont ajoutées indépendamment du comportement du processus, elles peuvent alors être inconsistantes avec la dynamique du système. L'automate obtenu est appelé automate contraint.
- 4- Application d'une analyse d'atteignabilité en avant de l'automate contraint pour calculer ses espaces d'état atteignables. Ces derniers sont réduits par rapport à ceux déterminés par l'analyse en avant de l'automate non contraint. Certains espaces peuvent aussi être supprimés, signifiant que les contraintes sont plus fortes que la dynamique du sommet correspondant.
- 5- Application d'une procédure amont/aval pour la détermination de la nouvelle garde de transition entre deux sommets.

Le chapitre suivant sera consacré au développement global de notre approche de commande maximale permissive. On y présentera un algorithme général de synthèse de la commande par le calcul des nouvelles gardes de transitions pour l'automate complet.

# Chapitre 4

## Synthèse de la commande dans le cas général

### 4.1 Introduction

Le chapitre précédent a présenté l'approche de calcul de la commande maximale permissive pour un sommet donné d'un automate hybride rectangulaire (AHR).

Nous nous intéressons ici à la généralisation de cette approche pour l'automate global. Nous allons ainsi proposer un algorithme général de synthèse de la commande (Batis et Alla, 2013a ; b ; c). Il tient compte d'une structure globale de l'automate, c'est-à-dire pouvant posséder des sommets à plusieurs entrées et plusieurs sorties, et contenant des branches et/ou des boucles. La procédure faite hors-ligne, mènera à la détermination optimale de l'ensemble des nouvelles gardes de transition de l'automate respectant les spécifications de commande imposées.

Comme nous l'avons expliqué dans le chapitre précédent, on s'intéresse à une commande temporisée du système (Section 3.2.4); c'est pour cela que nous allons montrer ensuite la procédure de calcul de l'horloge de commande. Celle-ci permet d'obtenir le superviseur temporisé qui agit sur le système en autorisant les événements contrôlables définis par un intervalle de temps. On aboutira alors à un algorithme englobant le calcul des nouvelles

gardes de transition dans un premier temps et de l'horloge de commande dans un deuxième temps.

Ce chapitre est organisé comme suit : d'abord, nous développons l'algorithme global de commande maximale permissive. Ensuite, nous présentons notre démarche de calcul de l'horloge de commande menant au superviseur temporisé. Puis, nous traiterons deux cas d'étude. Nous réserverons la dernière partie de ce chapitre à une extension de la procédure de commande au cas d'événements incontrôlables.

## 4.2 Procédure globale de commande

Le théorème 3.2 de commande maximale permissive développé dans le chapitre précédent, mène à la détermination de la nouvelle garde de transition  $V_{i-1}$  entre deux sommets  $(q_{i-1})$  et  $(q_{i(i \in [1, M])})$ . Son expression (figure 4.1) fait intervenir la garde initiale  $U_{i-1}$  contenant la contrainte ajoutée, l'espace d'état  $E_{i-1}$  du sommet  $(q_{i-1})$  et la nouvelle garde  $V_i$  à l'aval du sommet  $(q_i)$ .

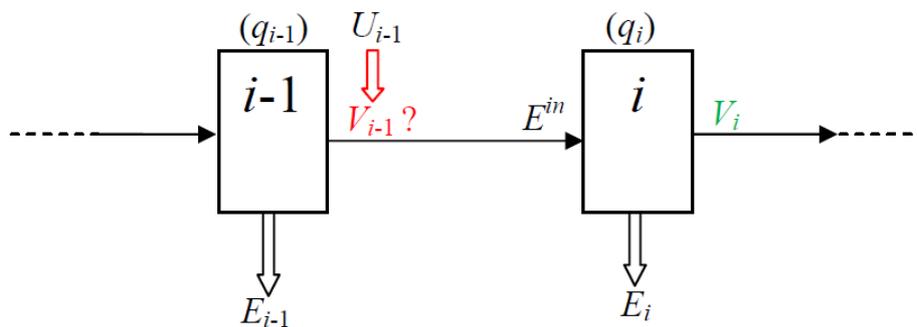


Fig.4.1- Calcul de la nouvelle garde  $V_{i-1}$ .

On s'intéresse à présent à généraliser cette approche à l'automate global. Ce dernier peut avoir une structure complexe. En effet, il peut être composé de branches et/ou de boucles. Il peut aussi avoir des sommets possédant plusieurs entrées et/ou plusieurs sorties. Le cas de plusieurs entrées ne change rien à l'expression de la nouvelle garde, puisque chaque branche sera traitée séparément. Par contre, le cas de plusieurs sorties est plus complexe. C'est pour cela que l'on va le traiter avant de développer l'algorithme global de commande.

### 4.2.1 Généralisation de l'approche au cas de sommets « multi-sorties »

Soit la partie d'un automate donnée par la figure 4.2 où le sommet  $(q_i)$  de l'automate contraint atteignable possède  $m$  transitions de sortie.

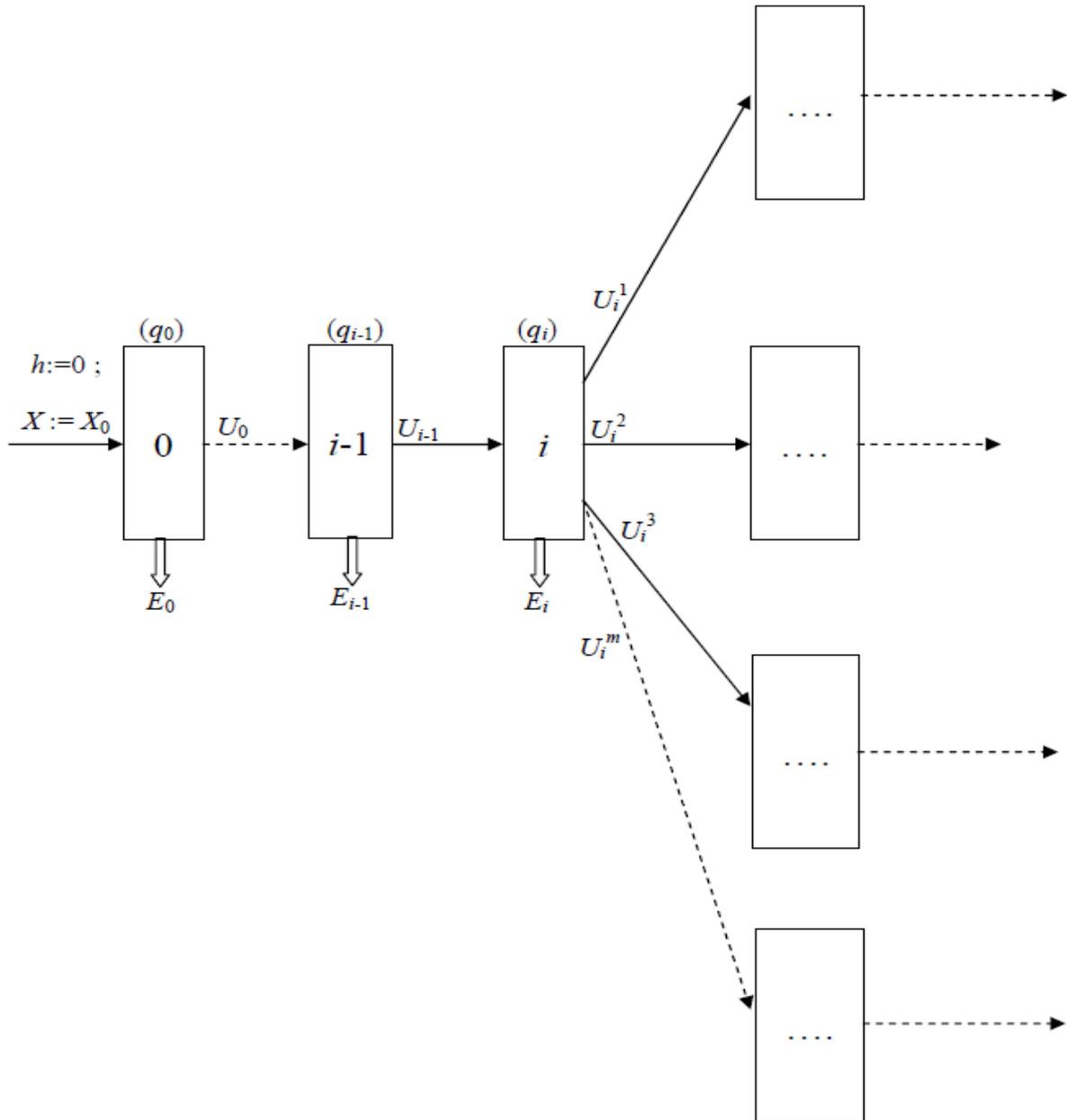


Fig.4.2- Partie d'un automate avec un sommet ayant plus d'une transition de sortie.

On notera par  $U_i^k (k \in [1, m])$  les gardes initiales des transitions de sortie du sommet  $(q_i)$ . On rappelle les principales variables pour une transition donnée  $k$  à la sortie de ce sommet (figure 4.3).

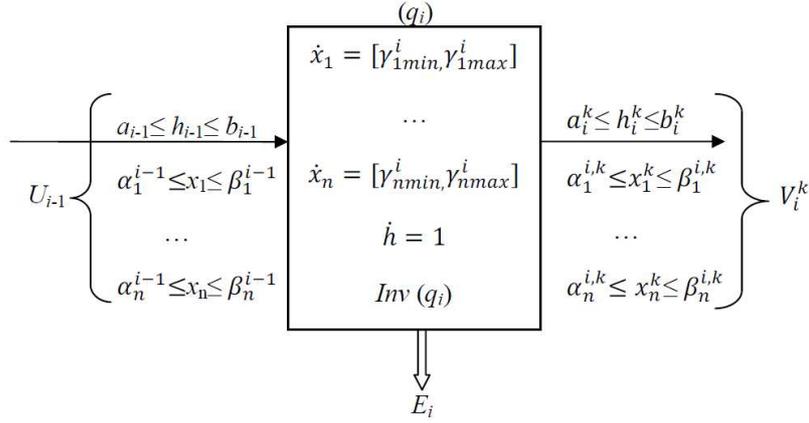


Fig.4.3- Principales variables pour une transition  $k$  donnée à la sortie du sommet  $(q_i)$ .

**Proposition 4.1. Cas d'un sommet ayant plus d'une transition de sortie**

Soient  $V_i^k (k \in [1, m])$  les nouvelles gardes calculées à l'aval du sommet  $(q_i)$ , alors la nouvelle garde  $V_{i-1}$  à la sortie de sommet  $(q_{i-1})$  est déterminée comme suit:

- Cas où  $\gamma_j^i > 0: V_{i-1} = E_{i-1} \wedge U_{i-1} \wedge (\bigwedge_{k=1}^m (\bigwedge_{j=1}^n [0, \beta_j^{i,k} - \gamma_{jmax}^i (a_i^k - b_{i-1}^k)]))$  (5)
- Cas où  $\gamma_j^i < 0: V_{i-1} = E_{i-1} \wedge U_{i-1} \wedge (\bigwedge_{k=1}^m (\bigwedge_{j=1}^n [\alpha_j^{i,k} + \gamma_{jmin}^i (a_i^k - b_{i-1}^k), \infty]))$  (6)

□

**Preuve :**

La proposition 4.1 est une généralisation directe du théorème 3.2. Sachant que le sommet  $(q_i)$  possède  $m$  transitions de sortie et que la nouvelle garde à l'amont de ce sommet dépend de l'aval, alors  $V_{i-1}$  doit respecter toutes les gardes à la sortie du sommet  $(q_i)$ . D'où l'ajout du terme de conjonction «  $\bigwedge_{k=1}^m (...)$  » dans les expressions (5) et (6) pour signifier qu'il s'agit de l'ensemble des gardes en sortie du sommet aval.

□

**4.2.2 Algorithme de synthèse de commande maximale permissive**

Notre objectif est de déterminer l'ensemble des nouvelles gardes de transition d'un automate global. La structure de l'automate des nouvelles gardes obtenu hors ligne est exactement la même que celle de l'automate contraint atteignable construit dans le chapitre 3.

Après avoir réalisé l'analyse d'atteignabilité en avant de l'automate contraint et construit l'automate contraint atteignable, la procédure de commande hors-ligne commence par le choix d'un sommet de départ dont on déterminera la nouvelle garde à son aval (Proposition 4.1). Celle-ci peut être remise en question après itération de l'approche aux sommets en amont. La procédure est répétée jusqu'à ce que tous les espaces atteignables d'entrée pour chaque sommet convergent vers une région fixe.

**Remarque 4.1.** Lorsque l'algorithme termine, ceci exprime le fait qu'il y ait une solution aux spécifications de commande désirées. Cependant, en général, la terminaison de l'algorithme n'est pas garantie. Fort heureusement, pour la plupart des systèmes réels, l'algorithme donne une solution. Trouver des conditions de terminaison nécessaires ou suffisantes dans le cas général représente une perspective importante pour notre travail de recherche.

□

L'algorithme ci-dessous est donné pour le cas d'un automate ayant  $(N+1)$  sommets notés de  $(q_0)$  à  $(q_N)$ . Le choix du sommet initial est un problème qui n'est pas traité en détail ici. Il dépend fortement de la structure du graphe de l'automate.

### Algorithme 1 : Procédure générale de commande hors-ligne

#### Etape 1 :

- Construction de l'automate hybride rectangulaire du système
- Ajout des spécifications de commande désirées  $\Rightarrow$  Automate contraint
- Analyse d'atteignabilité en avant de l'automate contraint et construction de l'automate contraint atteignable noté  $A$
- Construction de l'automate des nouvelles gardes noté  $A'$  ayant la même structure que  $A$  mais avec les dynamiques des variables dans ses sommets

#### Etape 2:

- $a = -1$  (Variable de vérification)
- **Tant que**  $(a < 0)$ 
  - {
  - Pour** tous les sommets  $(q_i)$

Calculer la nouvelle garde correspondante  $(V_i)^A$  en appliquant la Prop.4.1.

$(V_i)^{A''} \leftarrow (V_i)^A$  ( $A''$  est un automate pour sauvegarder les valeurs calculés)

**Fin Pour**

Choix d'un sommet ( $q_i$ )

Calculer la nouvelle garde correspondante  $(V_i)^A$  en appliquant la Prop.4.1.

**Si**  $(V_i)^A = (V_i)^{A''}$  (espace d'entrée du sommet inchangé) **alors**

{  $a = 1$

**Pour** tous les sommets ( $q_i$ )

$(V_i)^{A'} \leftarrow (V_i)^{A''}$

**Fin Pour**

}

**Sinon**  $a = -1$

**Fin Si**

}

### 4.2.3 Application au système tronçon de circulation

Dans le chapitre précédent, nous avons introduit le système tronçon de circulation, dont la variable d'état était le nombre de voitures dans le tronçon. L'automate contraint atteignable obtenu pour ce système est redonné par la figure 4.4.

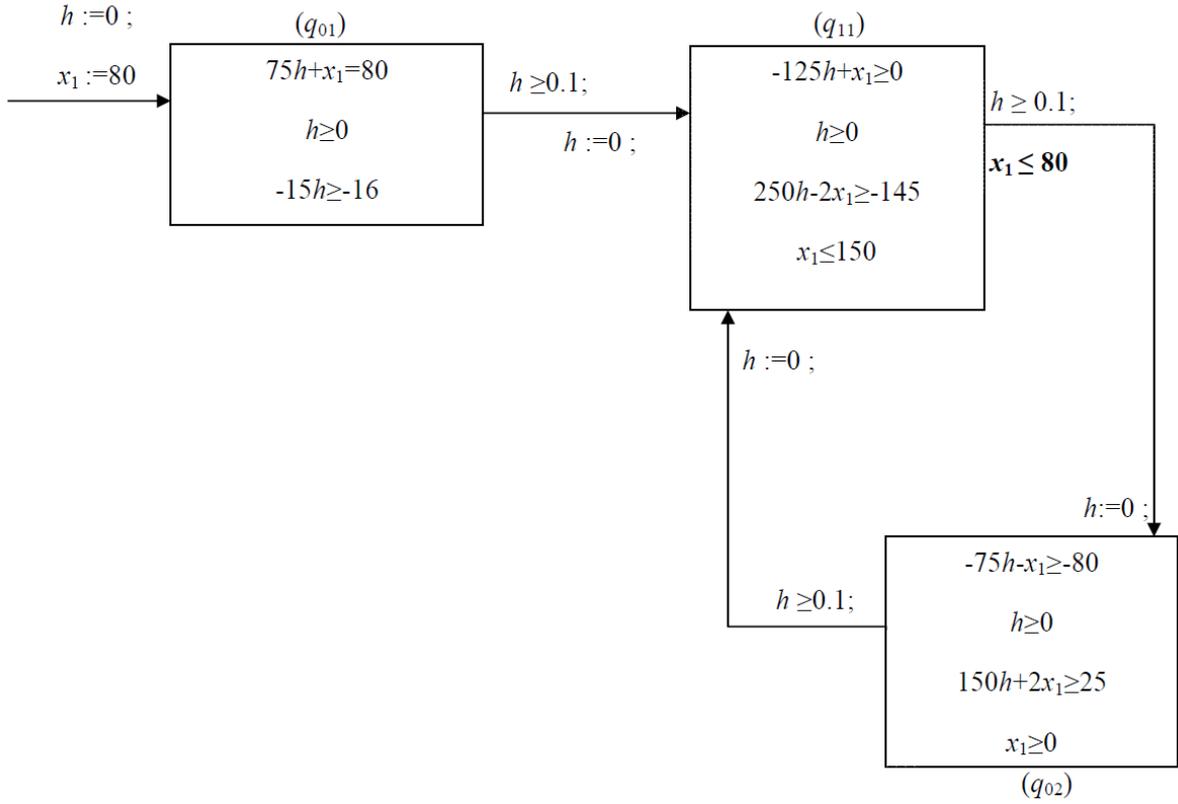


Fig.4.4- Automate contraint atteignable du système tronçon de circulation.

Dans cet automate, on a une boucle reliant les sommets  $(q_{11})$  et  $(q_{02})$  et une branche. Le calcul des nouvelles gardes commence au niveau de la boucle. Dans ce cas :

- Choix du sommet  $(q_{02})$ :  

$$V_{02} = E_{02} \wedge U_{02} \wedge [0, \beta_1^{11} - \gamma_1^{11}(a_{11} - b_{02})]$$

$$V_{02} = \{-75h - x_1 \geq -80 ; h \geq 0 ; 150h + 2x_1 \geq 25 ; x_1 \geq 0\} \wedge \{ h \geq 0.1\} \wedge \{0 \leq x_1 \leq 67.5\}$$
- On remonte  $\Rightarrow$  Sommet  $(q_{11})$   

$$V_{11} = E_{11} \wedge U_{11} \wedge [0, \beta_1^{02} - \gamma_1^{02}(a_{02} - b_{11})]$$

$$V_{11} = \{-125h + x_1 \geq 0 ; h \geq 0 ; 250h - 2x_1 \geq -145 ; x_1 \leq 150\} \wedge \{ h \geq 0.1 ; x_1 \leq 80\} \wedge \{x_1 \geq 7.5\}$$
- On remonte dans la boucle  $\Rightarrow$  Sommet  $(q_{02}) \Rightarrow$  On recalcule  $V_{02}$   

$$V_{02} = E_{02} \wedge U_{02} \wedge [0, \beta_1^{11} - \gamma_1^{11}(a_{11} - b_{02})]$$

$$V_{02} = \{-75h - x_1 \geq -80 ; h \geq 0 ; 150h + 2x_1 \geq 25 ; x_1 \geq 0\} \wedge \{ h \geq 0.1\} \wedge \{0 \leq x_1 \leq 67.5\}$$

$\Rightarrow$  On obtient le même résultat qu'au point de départ

$\Rightarrow$  On sort de la boucle
- Sommet  $(q_{01})$   

$$V_{01} = E_{01} \wedge U_{01} \wedge [0, \beta_1^{11} - \gamma_1^{11}(a_{11} - b_{01})]$$

$$V_{01} = \{75h + x_1 = 80 ; h \geq 0 ; 15h \leq 16\} \wedge \{h \geq 0.1\} \wedge \{0 \leq x_1 \leq 67.5\}$$

Finalement, on obtient le superviseur optimal donné par la figure 4.5.

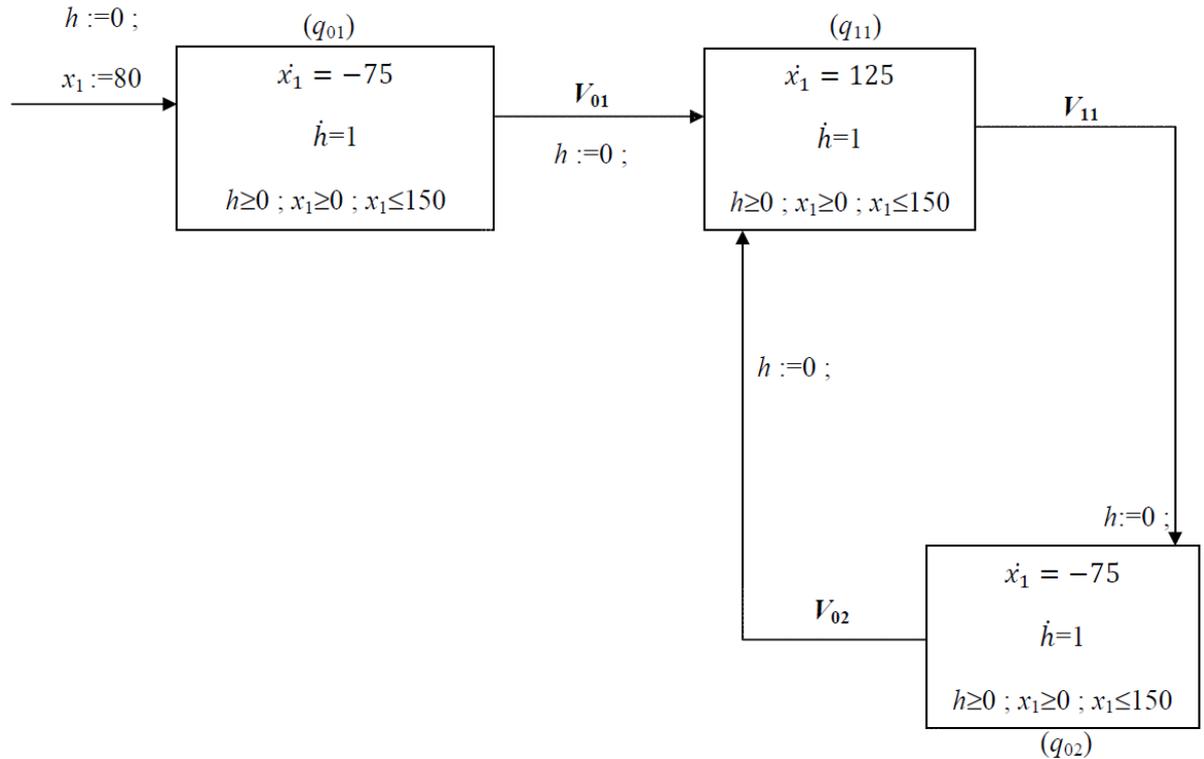


Fig.4.5- *Superviseur optimal du système tronçon de circulation.*

**Remarque 4.2.** L'algorithme termine pour cet exemple, donnant le superviseur optimal respectant les spécifications.

□

### 4.3 Commande temporisée maximale permissive

Comme nous l'avons expliqué dans le chapitre précédent (section 3.2.4), on s'intéresse à la réalisation d'une commande temporisée du système. En effet, seuls les événements discrets sur les gardes des transitions sont contrôlables. On peut alors modifier leur date d'occurrence dans leur intervalle d'existence.

Notre objectif final est de construire le superviseur temporisé du système (Batis et Alla, 2012a; 2012b); c'est pour cela que nous allons montrer la procédure de calcul de l'horloge de commande. Nous réaliserons en fin de compte la boucle de la figure 4.6, qui fonctionne de la

façon suivante : le système envoie l'événement contrôlable au superviseur, qui lui répond en lui indiquant l'intervalle d'occurrence où l'événement est autorisé.

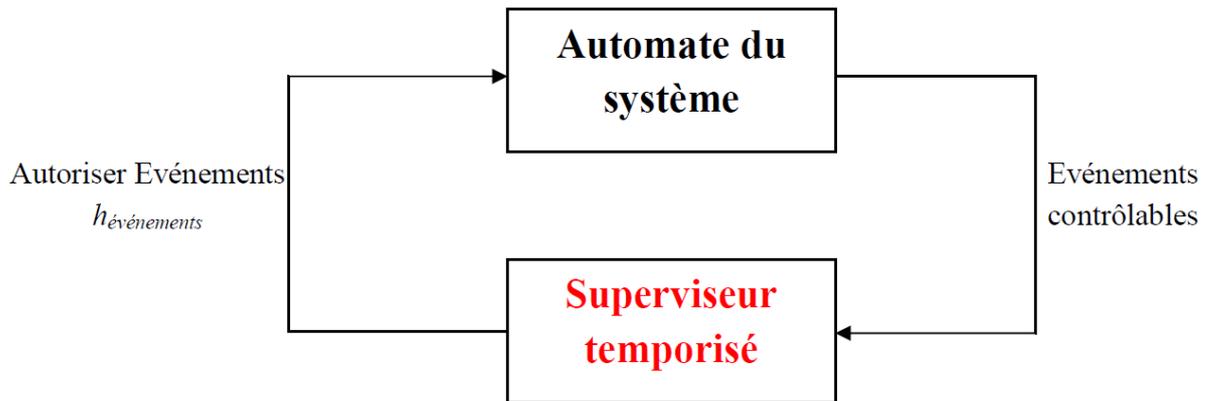


Fig.4.6- Boucle générale de commande.

L'horloge est un élément important dans la commande du système. Elle donne la date à laquelle la commutation d'un sommet à un autre doit avoir lieu. Elle correspond aux dates d'occurrence des événements contrôlables. A cause des calculs précédents (théorèmes 3.1, 3.2 et proposition 4.1), les valeurs d'horloge ont été modifiées et souvent réduites. Il est alors nécessaire de calculer les nouveaux intervalles d'occurrence des événements contrôlables, ce qui nous ramène à déterminer les intervalles définis dans la remarque 3.2.

**Proposition 4.2. Intervalles de commutation d'horloge**

L'intervalle de commutation de l'horloge dans la transition en aval du sommet ( $q_i$ ) est  $[\alpha_h^i, \beta_h^i]$ , tel que :

- $\alpha_h^i = \text{Minimum } h$
  - $\beta_h^i = \text{Maximum } h$
- } respectant : nouvelle garde  $V_i$

□

Les limites de l'intervalle d'horloge sont calculées par programmation linéaire (Matlab) respectant les inégalités qui définissent la nouvelle garde  $V_i$  déterminée dans les théorèmes 3.1

et 3.2 et la proposition 4.2. L'intervalle  $[\alpha_h^i, \beta_h^i]$  n'est jamais vide car le sommet  $(q_i)$  est atteignable dans l'automate contraint atteignable.

Revenons à l'exemple du tronçon de circulation dont nous avons calculé la nouvelle garde  $V_{01}$  entre les états  $(q_{01})$  et  $(q_{11})$ . Le calcul de l'intervalle d'horloge de commande  $[\alpha_h^{01}, \beta_h^{01}]$  se fait dans ce cas, par programmation linéaire, comme suit :

$$\left. \begin{array}{l} \bullet \alpha_h^{01} = \text{Minimum } h \\ \bullet \beta_h^{01} = \text{Maximum } h \end{array} \right\} \text{respectant : } V_{01} = \{75h + x_1 = 80 ; h \geq 0 ; 15h \leq 16\} \wedge \{h \geq 0.1\} \wedge \{0 \leq x_1 \leq 67.5\}$$

On trouve:  $h \in [0.17; 1.07]$ . Cet intervalle a bien été réduit par l'ajout des spécifications de commande. La valeur minimale (0.17 min) exprime le fait que le système doit rester un certain temps dans le sommet  $(q_{01})$  afin de respecter la contrainte ajoutée ( $x_1 \leq 67.5$ ) comme expliqué par la figure 3.14 du chapitre précédent. La valeur maximale (1.07 min), quant à elle, correspond au temps maximal de séjour dans le sommet correspondant, au-delà duquel le nombre de voitures devient négatif, ce qui est impossible.

Notre objectif est de construire le superviseur temporisé d'un système donné modélisé par un automate hybride rectangulaire. Nous avons déterminé d'une manière maximale permissive les nouvelles gardes de transition d'un automate, et calculé l'horloge de commande en utilisant ces nouvelles gardes. A présent, on s'intéresse à développer l'algorithme général donnant le superviseur temporisé maximal permissif du système.

La structure de l'automate de commande temporisé obtenu hors ligne est exactement la même que celle de l'automate contraint atteignable construit dans le chapitre 3. Afin d'obtenir le superviseur temporisé, on vide chaque sommet et chaque transition de l'automate contraint atteignable. On conserve uniquement les variables observables et contrôlables, c'est-à-dire l'horloge.

La procédure donnant le superviseur temporisé est dérivée de l'algorithme de commande Algorithme 1 (section 4.2.2). A partir de la nouvelle garde de transition, on calcule l'horloge de commande en appliquant la proposition 4.2.

**Algorithme 2 : Procédure de commande temporisée maximale permissive**

**Etape 1 :**

- Même pas que dans l'Algorithme 1
- Construction du modèle du superviseur temporisé noté  $C$  ayant la même structure que  $A$  mais ne contenant que la dynamique de l'horloge dans ses sommets

**Etape 2:**

- $a = -1$  (Variable de vérification)
- **Tant que** ( $a < 0$ )
  - {
  - Pour** Chaque sommet ( $q_i$ )
    - Calculer la nouvelle garde correspondante  $(V_i)^A$  en appliquant la Prop. 4.1.
    - Calculer la valeur de l'horloge de commande correspondante  $[\alpha_h^i, \beta_h^i]^A$  (Prop. 4.2)
    - $[\alpha_h^i, \beta_h^i]^{A'} \leftarrow [\alpha_h^i, \beta_h^i]^A$  ( $A'$  est un automate pour sauvegarder les valeurs calculés)
  - Fin Pour**
  - Choix d'un sommet ( $q_i$ )
  - Calculer la nouvelle garde correspondante  $(V_i)^A$  en appliquant le théorème 3.2.
  - Calculer la valeur de l'horloge de commande correspondante  $[\alpha_h^i, \beta_h^i]^A$  (Prop. 4.1)
  - Si**  $[\alpha_h^i, \beta_h^i]^{A'} = [\alpha_h^i, \beta_h^i]^A$  **alors**
    - { $a = 1$
    - Pour** Chaque sommet ( $q_i$ )
      - $[\alpha_h^i, \beta_h^i]^C \leftarrow [\alpha_h^i, \beta_h^i]^{A'}$
    - Fin Pour**
    - }
  - Sinon**  $a = -1$
  - Fin Si**
  - }

## 4.4 Etude de cas

Dans ce qui suit, nous allons traiter deux cas d'étude intéressants afin d'illustrer notre procédure de commande :

- Le premier cas concerne un système à deux tronçons de circulation. L'étude de ce système va nous permettre de voir comment traiter la forte interaction amont/aval dans une boucle.
- Le deuxième cas traite d'un système de production. Son intérêt réside dans l'étude du cas de sommets « multi-sorties ».

### 4.4.1 Système à deux tronçons de circulation

Le système de deux tronçons de circulation a été introduit dans le Chapitre 2 dans le cas des systèmes à événements discrets. Dans ce qui suit, on rappelle ce système, tout en y ajoutant des données afin de le traiter dans le cas hybride. En effet, nous avons vu que la structure du système n'était pas riche lorsque nous l'avons modélisé dans le cadre de systèmes à événements discrets. L'ajout de la dynamique des variables et de la contrainte temporelle va ainsi enrichir le modèle et par conséquent enrichir l'étude de la commande.

Le système est composé de deux tronçons avec des capacités et des vitesses de voitures différentes (figure 4.7).

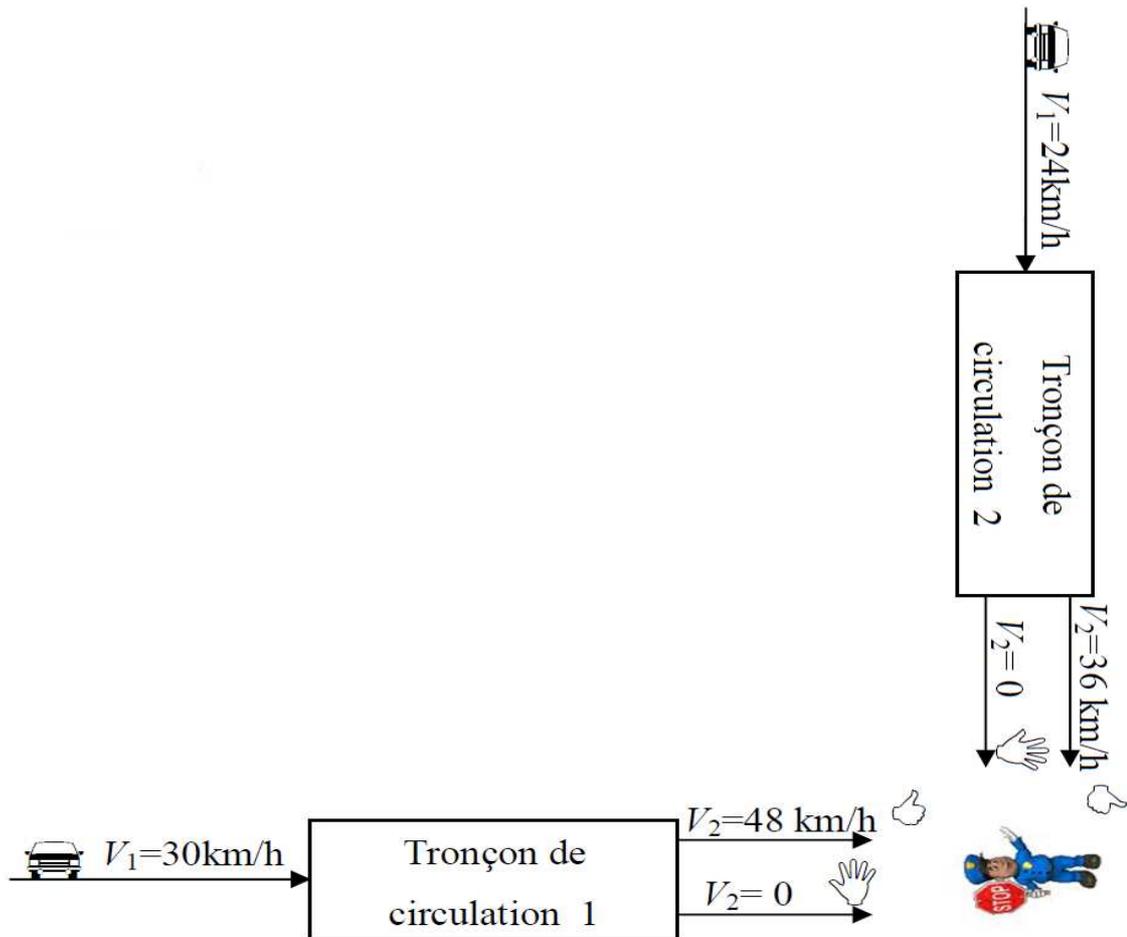


Fig.4.7- Système à 2 tronçons de circulation.

Les variables d'état sont les flux des voitures dans chaque tronçon  $x_1$  et  $x_2$ . Le tronçon de circulation 1 peut tolérer un nombre maximal de 150 voitures, alors que le tronçon 2 peut contenir jusqu'à 100 voitures. On suppose qu'au départ, le nombre de voitures dans chaque tronçon est  $x_1=80$  et  $x_2=10$ . On ajoute l'horloge  $h$  qui détermine les dates d'arrêt et de démarrage des voitures pour chaque tronçon dans un intervalle défini (événements contrôlables). Ici une seule horloge suffit car les états de marche et d'arrêt sont exclusifs. Cependant, dans le cas général, on peut être en présence de plusieurs horloges.

On souhaite ajouter les contraintes suivantes au système : un policier veut autoriser la circulation dans le tronçon 2 lorsque le nombre de voitures est  $x_2 \leq 90$  et dans le tronçon 1 lorsque le nombre de voitures est  $x_1 \leq 110$ .

L'automate contraint est donné par la figure 4.8.

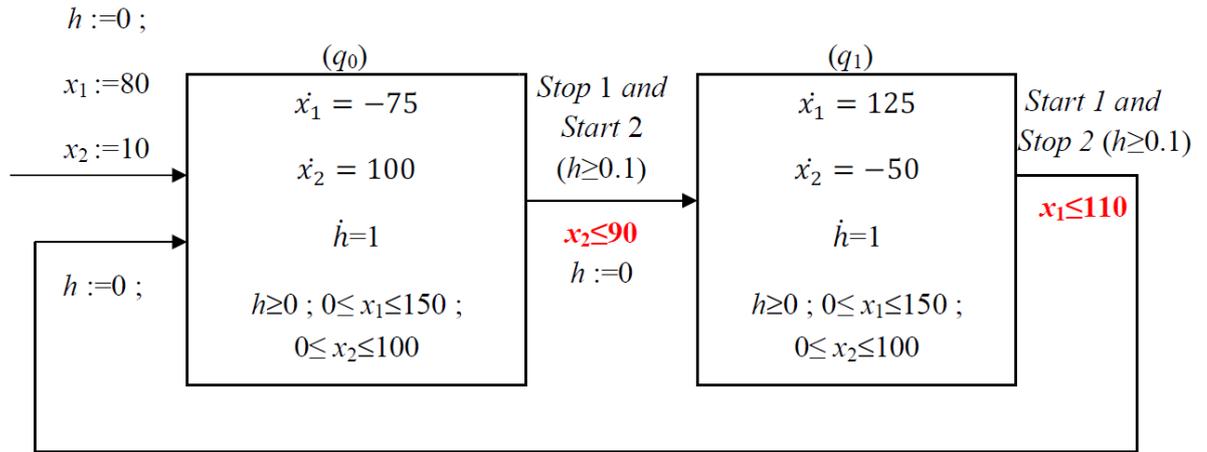


Fig.4.8- Automate contraint du système à 2 tronçons de circulation.

L'analyse d'atteignabilité en avant de cet automate contraint donne l'automate contraint atteignable à 8 sommets de la figure 4.9.

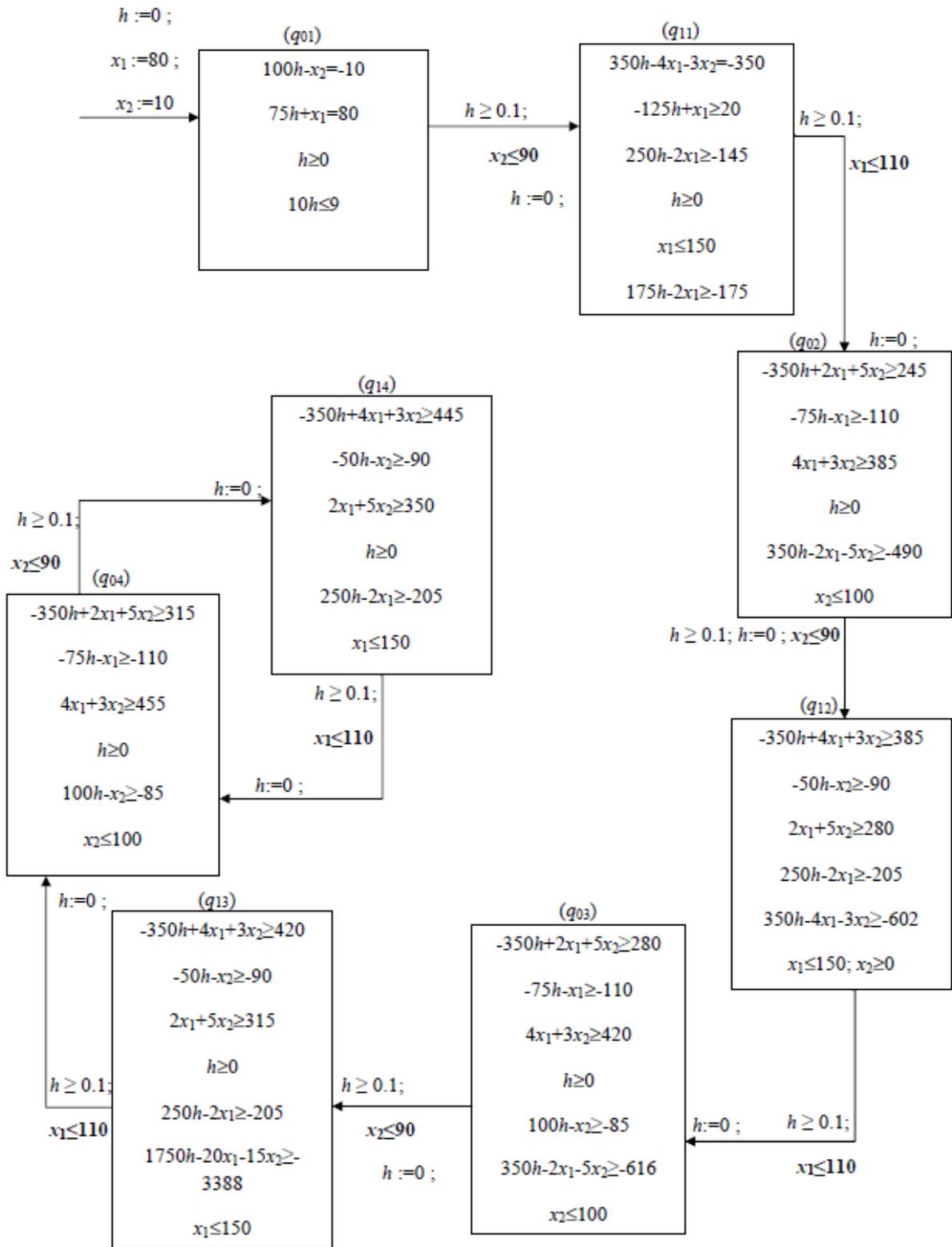


Fig.4.9- Automate constraintteignable du système à 2 tronçons de circulation.

On commence le calcul au niveau de la boucle. Le détail des différentes étapes est donné dans l'annexe A. On obtient dans ce cas les résultats importants suivants :

- Dans la boucle contenant les sommets  $(q_{04})$  et  $(q_{14})$ , il y a une forte interaction amont/aval (figure 4.10). Ceci prouve que le calcul de la commande n'est pas intuitif. On a du visiter la transition  $(q_{14}) \rightarrow (q_{04})$  deux fois afin de déterminer la commande maximale permissive.

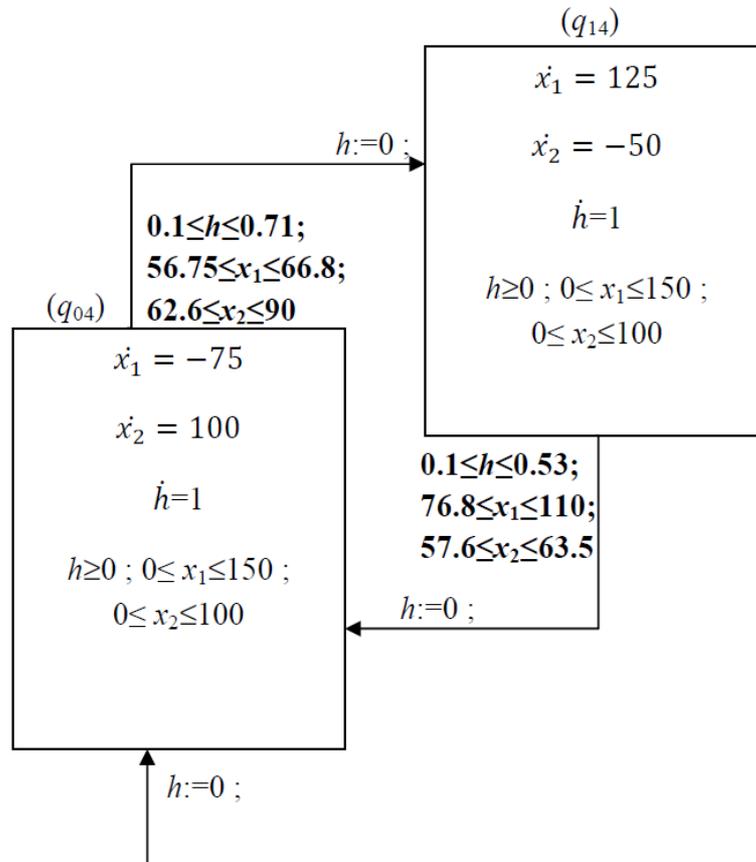


Fig.4.10- Forte interaction amont/aval.

- Concernant la transition  $(q_{01}) \rightarrow (q_{11})$ , la nouvelle garde aval  $V_{11}$  a fortement contraint la nouvelle garde amont de telle sorte que la nouvelle valeur d'horloge minimale a été réduite à  $h \geq 0.41$  min.
- L'algorithme termine pour cet exemple. Ceci exprime le fait qu'il y ait une solution aux spécifications de commande désirées.

On construit le superviseur temporel maximal permissif de ce système en vidant les sommets et les transitions de l'automate contraint atteignable tout en ne gardant que les valeurs calculées de l'horloge de commande. Le résultat est donné par la figure 4.11.

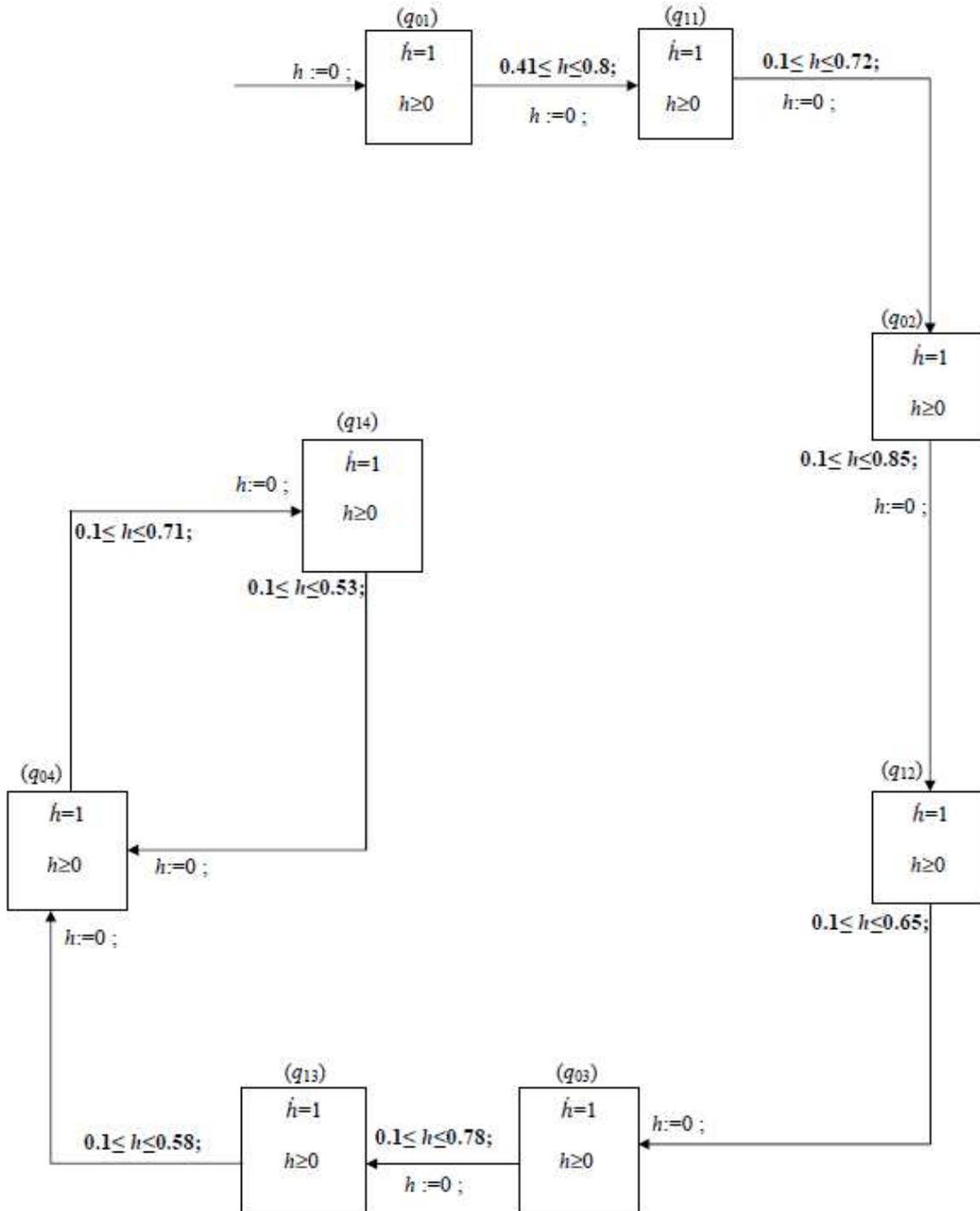


Fig.4.11- Superviseur temporel maximal permissif du système à 2 tronçons de circulation.

**Remarque 4.3.** La solution proposée ci-dessus correspond à la solution optimale, c'est-à-dire toutes les trajectoires possibles de l'automate contraint atteignable. Elle est non déterministe.

Il est évident que dans une commande en temps-réel, le contrôleur doit être déterministe. Le choix d'une valeur fixe de commutation pour la première transition après l'initialisation aura une influence restrictive sur la commande avale. Ceci revient à déterminer une trajectoire particulière parmi toutes les trajectoires possibles. On doit alors, à partir de l'algorithme de commande maximale permissive hors-ligne développé ci-dessus, déterminer une procédure de commande en ligne.

□

#### 4.4.2 Système de production

Soit un système de production composé d'un centre de production exclusive partagée qui alimente deux stocks destinés à la consommation ayant une capacité maximale de 100 éléments pour chaque stock. Le système est illustré par la figure 4.12.

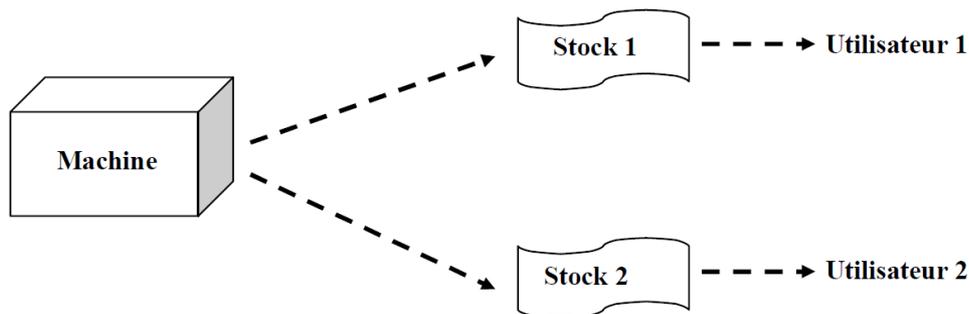


Fig.4.12- *Système de production exclusive partagée.*

Les variables d'état sont les quantités dans chaque stock  $x_1$  et  $x_2$ . On suppose qu'au départ, les stocks sont à moitié plein ( $x_1=x_2=50$ ). On ajoute à ces variables une horloge  $h$  qui détermine les dates de livraison dans un intervalle défini (événements contrôlables).

La machine est partagée mais d'une façon exclusive. Son taux de production est constant et égal à 5. D'autre part, la consommation fluctue durant la journée (Tableaux 4.1 et 4.2). Celle-ci n'est pas connue avec certitude. Ceci est modélisé dans l'AHR par des intervalles associés aux variables flux.

Le flux bilan varie avec le temps. La transition  $(q_i) \rightarrow (q_{i+1})$  a lieu suite à l'occurrence de l'événement contrôlable  $\sigma_i$ . L'automate correspondant (pour un comportement journalier normal) est donné par la figure 4.13, sans tenir compte des contraintes en gras.

Les changements d'état sont non déterministes, ce qui mène à un degré de liberté pour la commande. Par exemple, l'événement contrôlable  $\sigma_0$  correspond à une augmentation de consommation dans le stock 1 (le bilan passe de la valeur 4 à 3), qui peut avoir lieu pour une valeur de  $h \in [6, 8]$ .

Les états  $(q_0)$ ,  $(q_1)$  et  $(q_2)$  correspondent à la production dans le stock 1. Les états  $(q_3)$ ,  $(q_4)$ ,  $(q_5)$  et  $(q_5')$  correspondent à la production dans le stock 2. Les tableaux 4.1 et 4.2 résument les différentes valeurs de production et de consommation pour chaque stock.

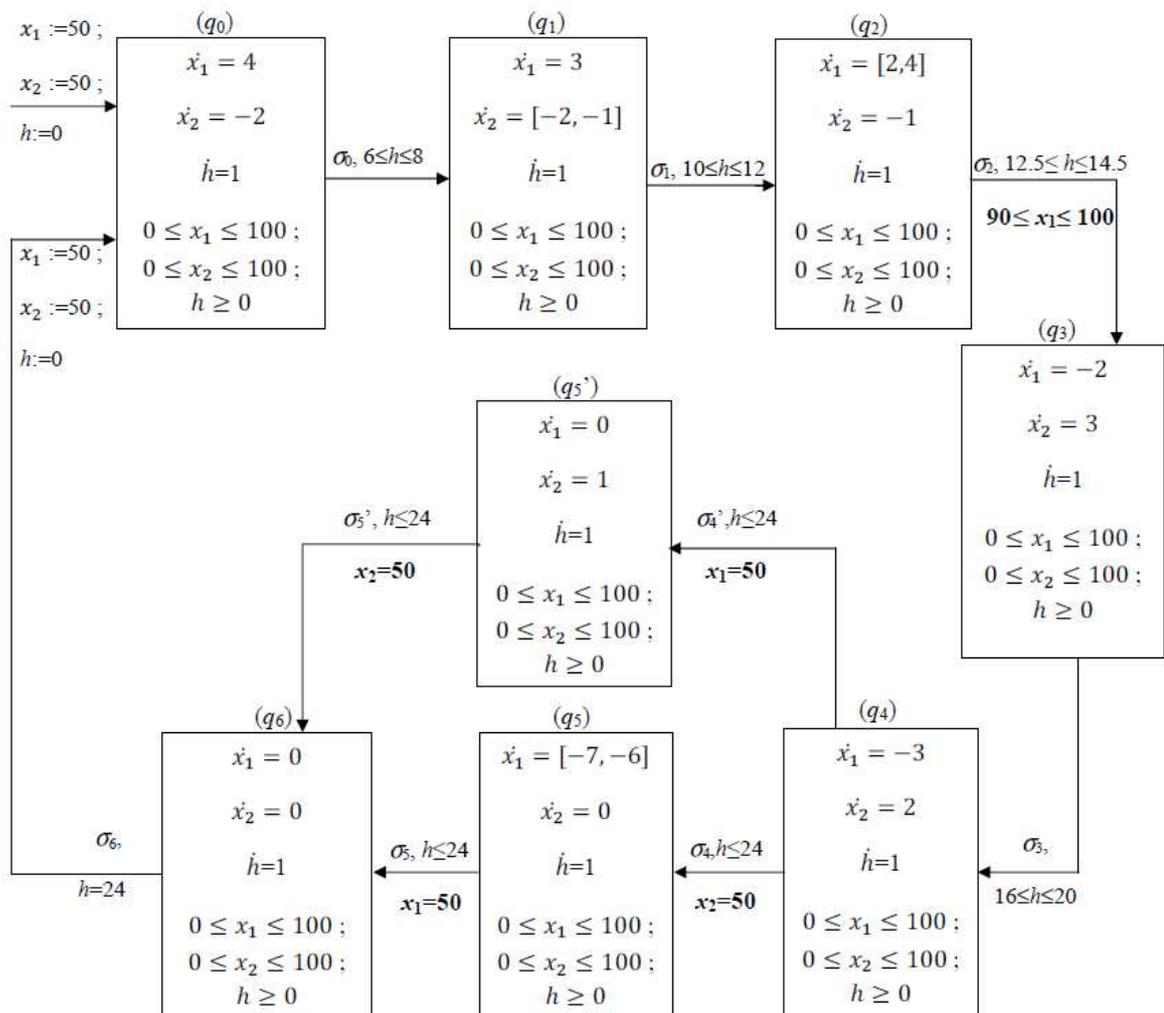


Fig.4.13- Le modèle automate du système de production.

On remarque que, à partir du sommet  $(q_4)$ , il y a deux chemins possibles. Ces deux branches expriment le fait qu'on ne connaît pas à priori l'ordre dans lequel les deux stocks vont se vider.

**Tableau 4.1. Valeurs de production et de consommation pour le stock 1**

Location	$(q_0)$	$(q_1)$	$(q_2)$	$(q_3)$	$(q_4)$	$(q_5)$	$(q_5')$	$(q_6)$
<b>Production</b>	5	5	5	0	0	0	0	0
<b>Consumption</b>	1	2	[1, 3]	2	3	[5 6]	0	0

**Tableau 4.2. Valeurs de production et de consommation pour le stock 2**

Location	$(q_0)$	$(q_1)$	$(q_2)$	$(q_3)$	$(q_4)$	$(q_5)$	$(q_5')$	$(q_6)$
<b>Production</b>	0	0	0	5	5	5	5	0
<b>Consumption</b>	2	[1 2]	1	2	3	5	4	0

On souhaite imposer les contraintes suivantes au système (elles sont ajoutées en gras dans le modèle automate de la figure 4.13) :

- Transition  $(q_2) \rightarrow (q_3)$  : on veut arrêter la production dans le stock 1 lorsque le niveau est  $x_1 \in [90, 100]$ .
- Transitions  $(q_4) \rightarrow (q_5)$ ,  $(q_5) \rightarrow (q_6)$ ,  $(q_4) \rightarrow (q_5')$  et  $(q_5') \rightarrow (q_6)$  : on veut qu'à la fin de la journée, les quantités dans les deux stocks retournent à leur valeurs initiales  $x_1=50$  et  $x_2=50$ .

Les autres transitions n'ont pas de contraintes. Afin de déterminer le superviseur temporel maximal permissif du système, on applique notre approche développée ci-dessus.

L'analyse en avant de l'automate contraint donne sept sommets menant à l'automate contraint atteignable donné par la figure 4.14. Ce dernier est à la fois réversible et non bloquant. Il existe alors un superviseur optimal respectant les spécifications.

On remarque que le sommet  $(q_5')$  a été supprimé par l'analyse en avant. Ceci signifie qu'il n'est pas atteignable par sa contrainte amont  $x_1=50$ .

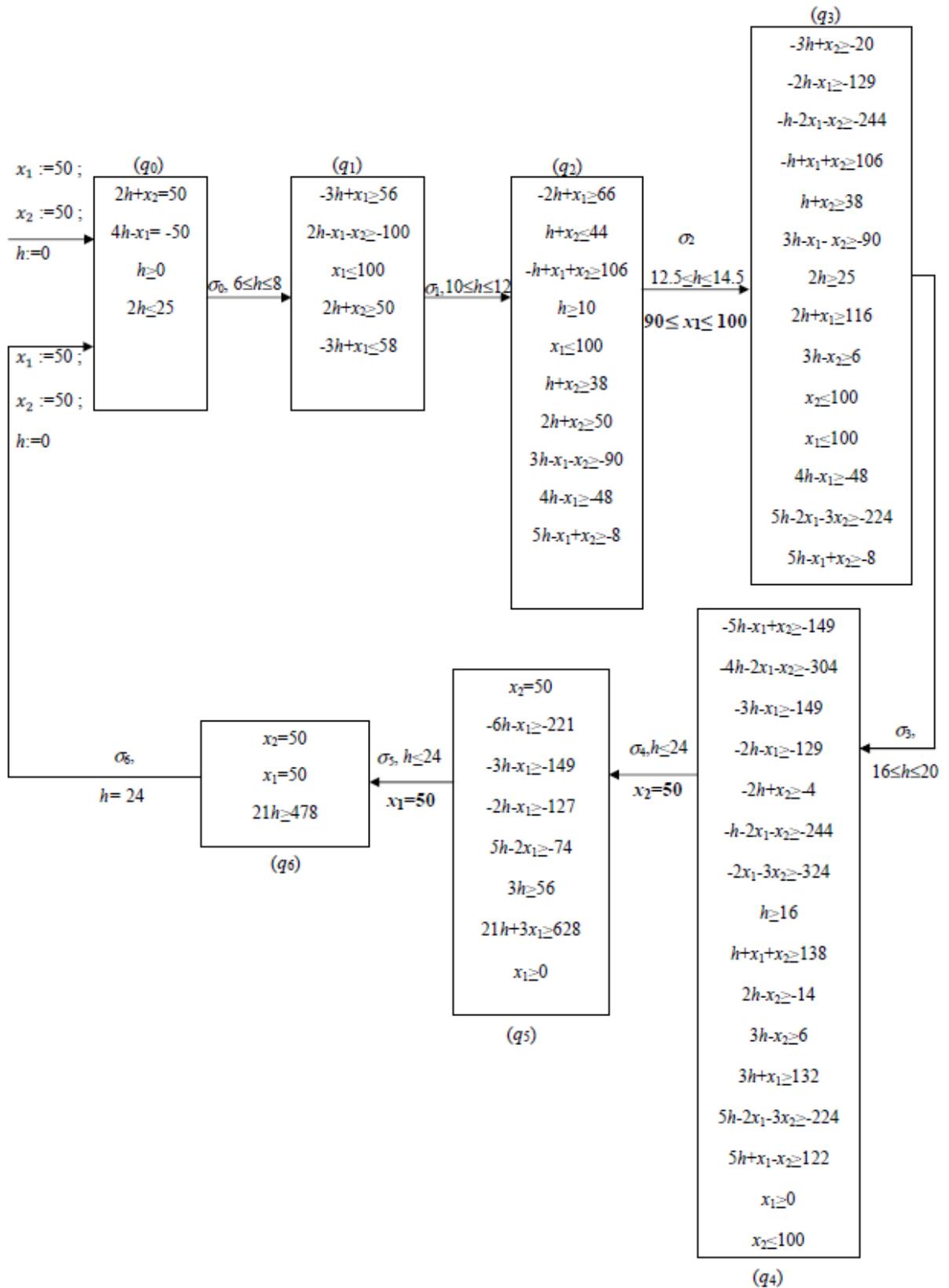


Fig.4.14- Automate constraintteignable du système de production.

On se retrouve alors dans le cas simple d'une boucle contenant des sommets avec une seule transition de sortie.

Le calcul du superviseur temporel maximal permissif se fait alors par application de la procédure développée dans la section 4.3. Le résultat est donné par la figure 4.15.

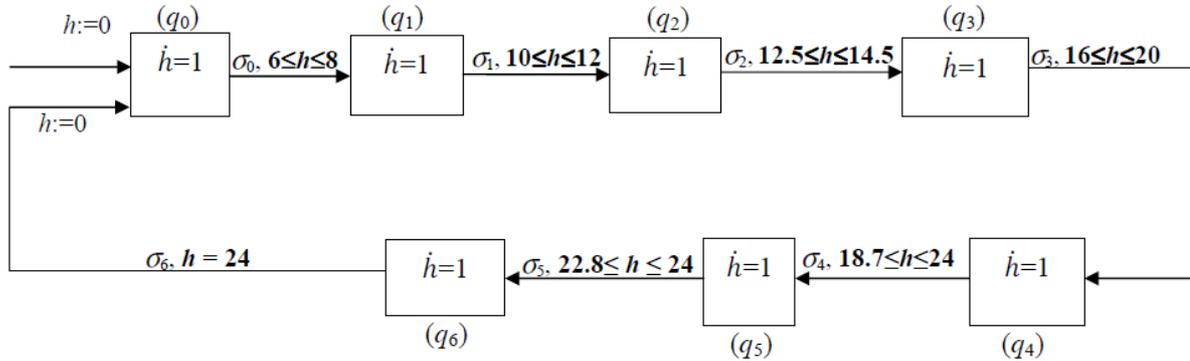


Fig.4.15- *Superviseur temporel maximal permissif du système de production.*

On constate que toutes les valeurs initiales de l'horloge sont atteignables, sauf pour les transitions contrôlables suivantes :

- $(q_4) \rightarrow (q_5)$  : l'intervalle d'horloge initiale ( $h \leq 24$ ) a été ramené à ( $18:35 \leq h \leq 24$ ). La limite inférieure 18:35 est due à la contrainte  $x_2=50$ . Ceci signifie que l'on doit changer de taux de consommation à partir de 18:35 afin de satisfaire le fait que le stock 2 soit à moitié plein.
- $(q_5) \rightarrow (q_6)$  : l'intervalle d'horloge initiale ( $h \leq 24$ ) a été ramené à ( $23:20 \leq h \leq 24$ ).

Pour illustrer le cas d'un sommet ayant plusieurs transitions de sortie, nous avons modifié le modèle automate de la façon suivante: on suppose que, à partir du sommet  $(q_1)$ , deux chemins sont possibles. Cette nouvelle configuration a été traitée en détail dans l'annexe B.

**Remarque 4.4.** Jusqu'à présent, notre objectif était de déterminer une commande maximale permissive donnant toutes les trajectoires possibles d'un système respectant les spécifications de commande qu'on lui a imposées. Le superviseur temporel qu'on obtient par cette approche hors-ligne est non déterministe, comme nous l'avons souligné précédemment. Afin de le rendre déterministe, on doit choisir une trajectoire particulière parmi toutes les trajectoires précédemment calculées. Ce calcul se fait de manière itérative. Si on fixe une valeur de l'horloge de commande dans son intervalle maximal permissif au niveau de la

première transition après l'initialisation, l'intervalle maximal permissif de l'horloge de commande dans la transition qui suit sera inévitablement réduit. Et ce sera logiquement dans ce nouvel intervalle que sera affectée la nouvelle valeur d'horloge, et ainsi de suite.

Considérons le système de production dont le modèle automate contraint est donné par la figure 4.13. Le calcul des nouvelles gardes par application de la procédure de commande maximale permissive de la section 4.2 donne pour la première transition après l'initialisation :

$$V_0 = \{6 \leq h \leq 8 ; 74 \leq x_1 \leq 82 ; 34 \leq x_2 \leq 38\}$$

On choisit comme trajectoire possible celle qui change le taux de consommation au plus tôt. Dans ce cas, la première transition doit être faite à 6 heures, ce qui correspond à une quantité de 74 dans le stock 1 et de 38 dans le stock 2. En fixant ces valeurs, on doit refaire une analyse d'atteignabilité en avant du nouvel automate obtenu. A partir de l'automate contraint atteignable de départ, on ajoute les nouvelles gardes obtenues par la procédure hors-ligne et les valeurs fixes de la première garde  $V_0' = \{h=6 ; x_1=74 ; x_2=38\}$ . La démarche recommence alors en déterminant les nouvelles gardes du nouvel automate atteignable et en fixant la valeur d'horloge sur la garde suivante dans le nouvel intervalle calculé.

L'algorithme donnant la procédure de commande en ligne est coûteux en calcul. En effet, à chaque étape, il y a une remise en cause de l'intervalle de commande. Nous devons alors refaire une analyse en avant chaque fois qu'une valeur d'horloge est fixée.

□

### 4.5 Extension de la procédure de commande au cas d'événements incontrôlables

Notre approche de commande maximale permissive est basée sur l'hypothèse que tous les événements du modèle automate hybride rectangulaire sont contrôlables. Nous avons récemment voulu étendre notre approche au cas d'événements incontrôlables. Nos travaux sont encore en cours de développement (Batis et Alla, 2013a).

Un événement est dit incontrôlable, lorsque l'on ne peut pas modifier sa date d'occurrence dans son intervalle d'existence.

A partir de cette affirmation, le calcul de la nouvelle garde pour une transition donnée menant à une commande maximale permissive se fait en appliquant la proposition 4.1, en prenant en compte le caractère contrôlable ou non de l'événement:

### Proposition 4.3. Extension au cas incontrôlable

Etant données la garde initiale ( $U_{i-1}$ ) et la nouvelle garde ( $V_{i-1}$ ) à la sortie du sommet ( $q_{i-1}$ ):

- Si l'événement correspondant  $\sigma_{i-1}$  est contrôlable, alors la nouvelle garde est ( $V_{i-1}$ ).
- Si l'événement correspondant  $\sigma_{i-1}$  est incontrôlable, alors le sommet aval ( $q_i$ ) devient interdit lorsque ( $V_{i-1} \neq U_{i-1}$ ).

□

La preuve de cette proposition est triviale car si l'événement est incontrôlable, alors on ne peut pas modifier son intervalle d'horloge de commande. Cela signifie que, pour qu'il puisse y avoir une solution au problème de commande, il faut que la nouvelle garde calculée soit la même que la garde initiale, i.e. ( $V_{i-1} = U_{i-1}$ ). Dans le cas contraire, le sommet suivant devient interdit rendant une partie du modèle non atteignable. Il peut arriver dans certains cas que la solution optimale au problème de commande soit l'ensemble vide.

Afin d'appliquer cette condition, nous nous sommes intéressés à un système d'approvisionnement en eau dont l'illustration générale est donnée par la figure 4.16.

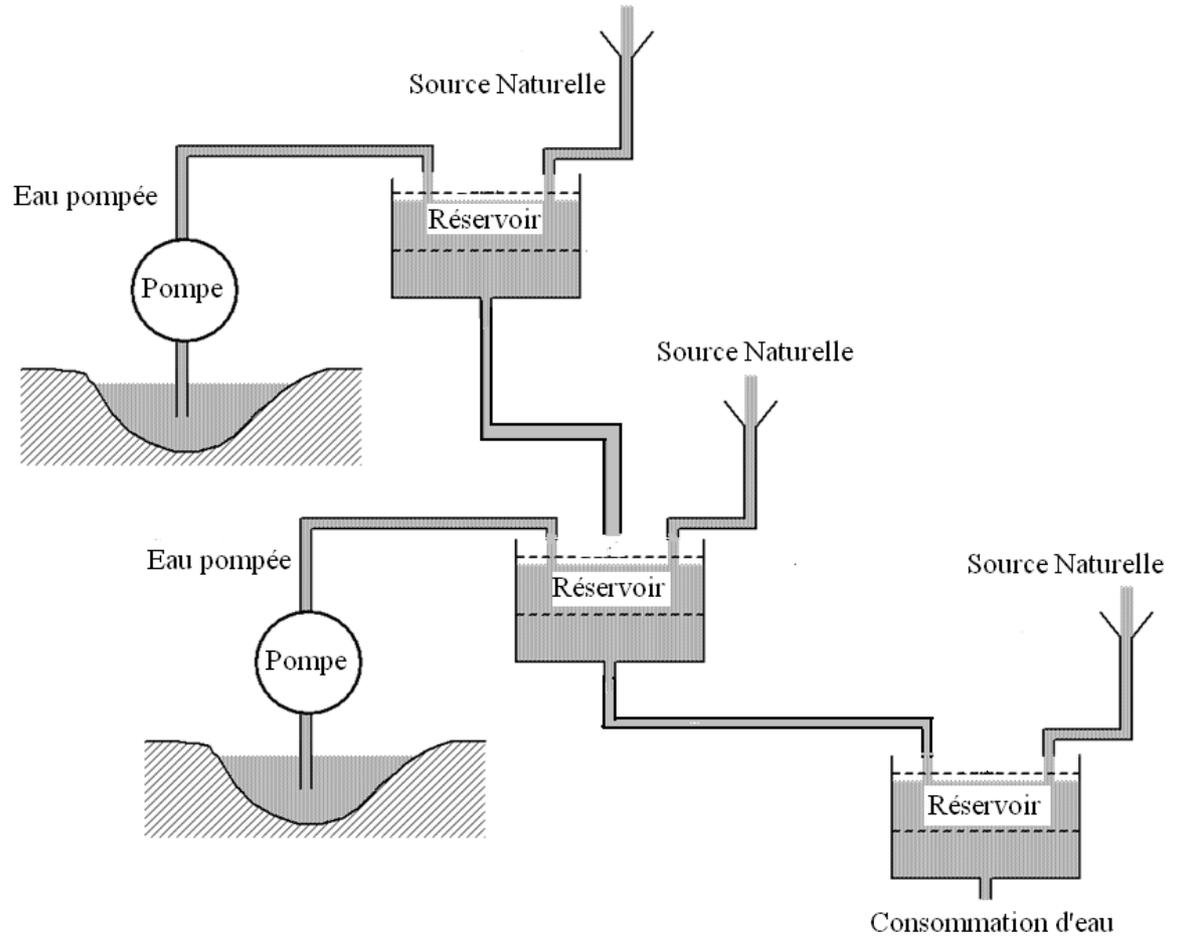


Fig.4.16- *Système général d'approvisionnement en eau.*

Ce système est composé d'un ensemble de réservoirs qui sont alimentés par l'eau provenant de la nappe phréatique, par les sources naturelles (pluie) ou par d'autres réservoirs. Les flux d'eau sont supposés constants puisque c'est une bonne approximation au niveau macroscopique dans lequel le système est étudié. On suppose aussi que les pertes de charge dans les tuyaux sont négligeables.

Le système est hybride car le flux d'eau est continu alors que les ouvertures et fermetures des pompes sont discrètes. L'approvisionnement par l'eau de pluie peut être modélisé par une fonction constante par morceaux. Les événements d'ouverture et de fermeture des pompes sont contrôlables car l'utilisateur peut contrôler leurs dates d'occurrence. Les autres événements relatifs à un changement dans le taux de consommation ou dans la source d'approvisionnement sont incontrôlables.

Les pompes utilisent l'énergie électrique dont le coût dépend de l'heure à laquelle elles sont utilisées. Par conséquent, la fonction coût devient un paramètre important pour la commande d'un tel système.

Les différentes variables d'état de ce système peuvent être :

- Le volume d'eau dans chaque réservoir
- La fonction coût des pompes

On ajoute à ces variables une horloge  $h$  qui détermine les dates de commutation de vannes.

Le problème pour ce type de système est que l'on doit avoir un certain niveau d'eau dans les réservoirs, à différents moments de la journée, dans le but de satisfaire les besoins en consommation avec un coût restreint. D'où l'intérêt de la commande qui consiste à déterminer les dates où les pompes doivent marcher ou s'arrêter.

Afin de simplifier l'application de notre approche, on choisit une portion du système comme le montre la figure 4.17. Notre approche peut être facilement généralisée au système général.

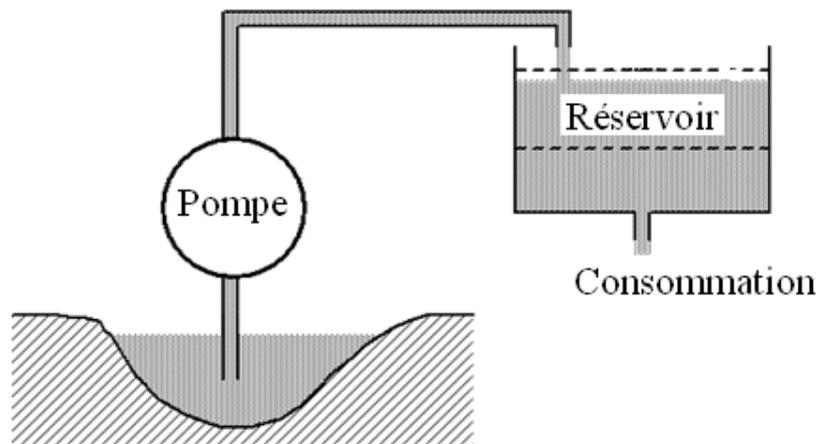


Fig.4.17- *Système d'approvisionnement en eau.*

Il s'agit d'un réservoir alimenté par l'eau provenant de la nappe phréatique. Son volume maximum est de  $70 \text{ m}^3$ . Les variables d'état sont :

- $x_1$  : le volume d'eau dans le réservoir ( $0 \leq x_1 \leq 70 \text{ m}^3$ )
- $x_2$  : la fonction coût ( $0 \leq x_2 \leq 100 \text{ kWh}$ ) qui correspond à une énergie dont la valeur maximale autorisée est de 100 kWh.

Les intervalles entre parenthèses correspondent aux invariants du système qui doivent toujours être respectés. A ces variables, on ajoute l'horloge  $h$  qui détermine les dates de commutation des vannes.

L'automate de la figure 4.18 décrit l'évolution des variables pour un comportement journalier normal (sans prendre en considération les expressions en gras au niveau des transitions).

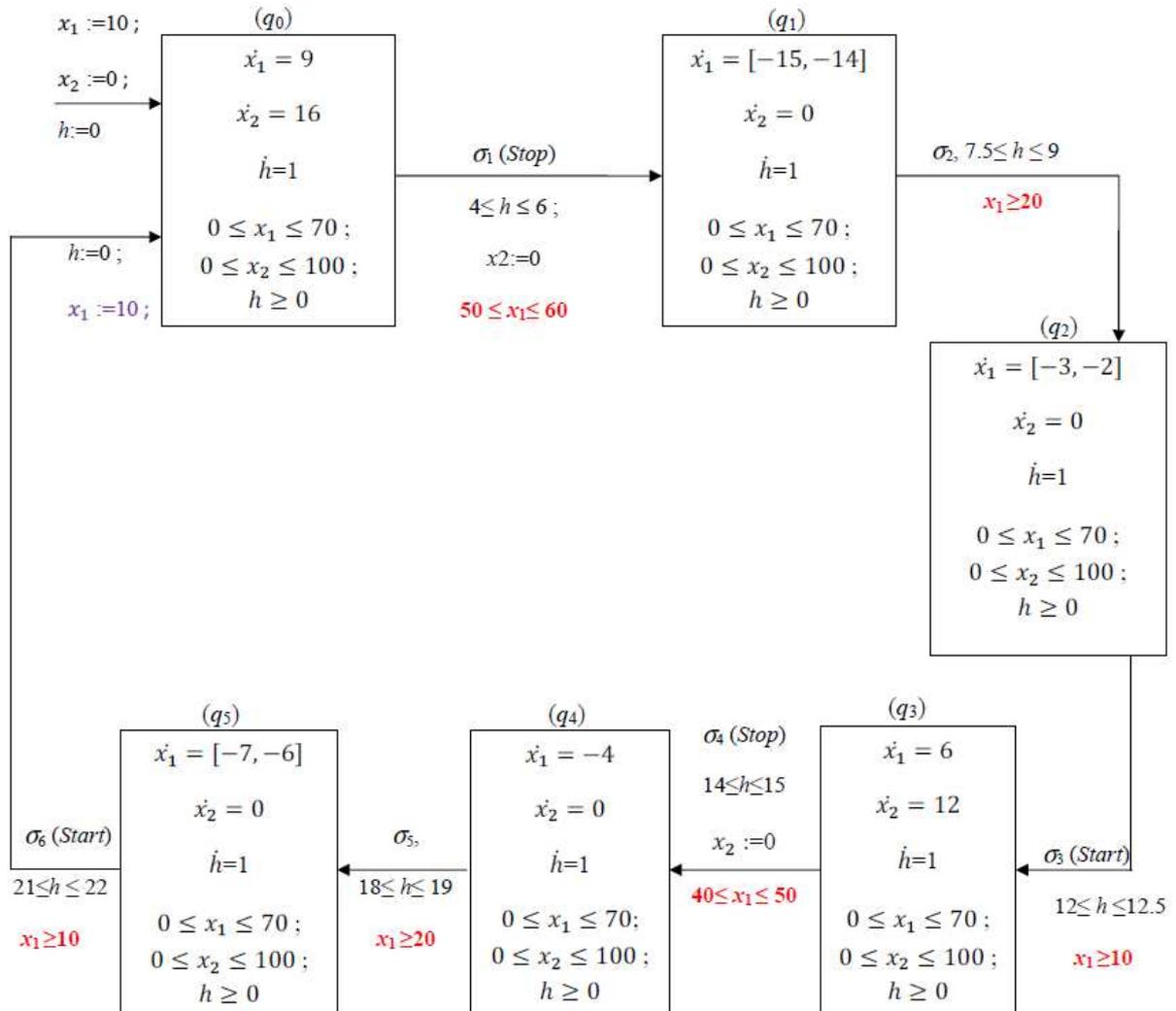


Fig.4.18- Le modèle automate du système d'approvisionnement en eau.

Le signe du flux de  $x_1$  est donné par la différence entre l'alimentation et la consommation en eau. Concernant la pente de  $x_2$ , elle est positive lorsque la pompe est en marche et nulle quant elle est à l'arrêt.

Au départ (sommet  $(q_0)$ ), la pompe est en marche à pleine puissance ( $\dot{x}_2=16$ ), la production augmente avec un taux de  $10\text{m}^3/\text{heure}$  et il y a une faible consommation de  $1\text{m}^3/\text{heure}$  ( $\dot{x}_1=9$ ). Puis, on arrête la pompe. A  $12 \leq h \leq 12:30$ , on redémarre la pompe mais à  $\frac{3}{4}$  de puissance ( $\dot{x}_2=12$ ). Dans ce cas, la production augmente avec un taux de  $7.5\text{m}^3/\text{heure}$  et il y a une faible consommation de  $1.5\text{m}^3/\text{heure}$  ( $\dot{x}_1=6$ ). Le tableau 4.3 résume les différentes valeurs de production et de consommation dans chaque sommet.

**Tableau 4.3. Valeurs de production, de consommation et de coût**

Location \ Flux	$(q_0)$	$(q_1)$	$(q_2)$	$(q_3)$	$(q_4)$	$(q_5)$
<b>Production (<math>\text{m}^3/\text{h}</math>)</b>	10	0	0	7.5	0	0
<b>Consumption (<math>\text{m}^3/\text{h}</math>)</b>	1	[14,15]	[2, 3]	1.5	4	[6 7]
<b>Coût(kW)</b>	16	0	0	12	0	0

Les événements contrôlables « *Start* » et « *Stop* » correspondent au démarrage et arrêt de la pompe. Les autres événements ( $\sigma_1$  et  $\sigma_4$ ), se référant à un changement dans le taux de consommation sont incontrôlables.

Afin de satisfaire les besoins en consommation avec un coût optimal, on ajoute les spécifications de commande suivante (figure 4.18) :

- Transition  $(q_0) \rightarrow (q_1)$ : on veut arrêter la pompe lorsque le volume est entre  $50$  et  $60 \text{ m}^3$
- Transition  $(q_3) \rightarrow (q_4)$ : on veut arrêter la pompe lorsque le volume est entre  $40$  et  $50 \text{ m}^3$
- Transitions  $(q_1) \rightarrow (q_2)$ ,  $(q_4) \rightarrow (q_5)$ : on veut que le volume dans le réservoir soit supérieur à  $20\text{m}^3$ .
- Transitions  $(q_2) \rightarrow (q_3)$ ,  $(q_5) \rightarrow (q_0)$ : on veut que le volume dans le réservoir soit supérieur à  $10\text{m}^3$ .

L'analyse en avant de l'automate contraint donne 6 sommets menant à l'automate contraint atteignable de la figure 4.19.

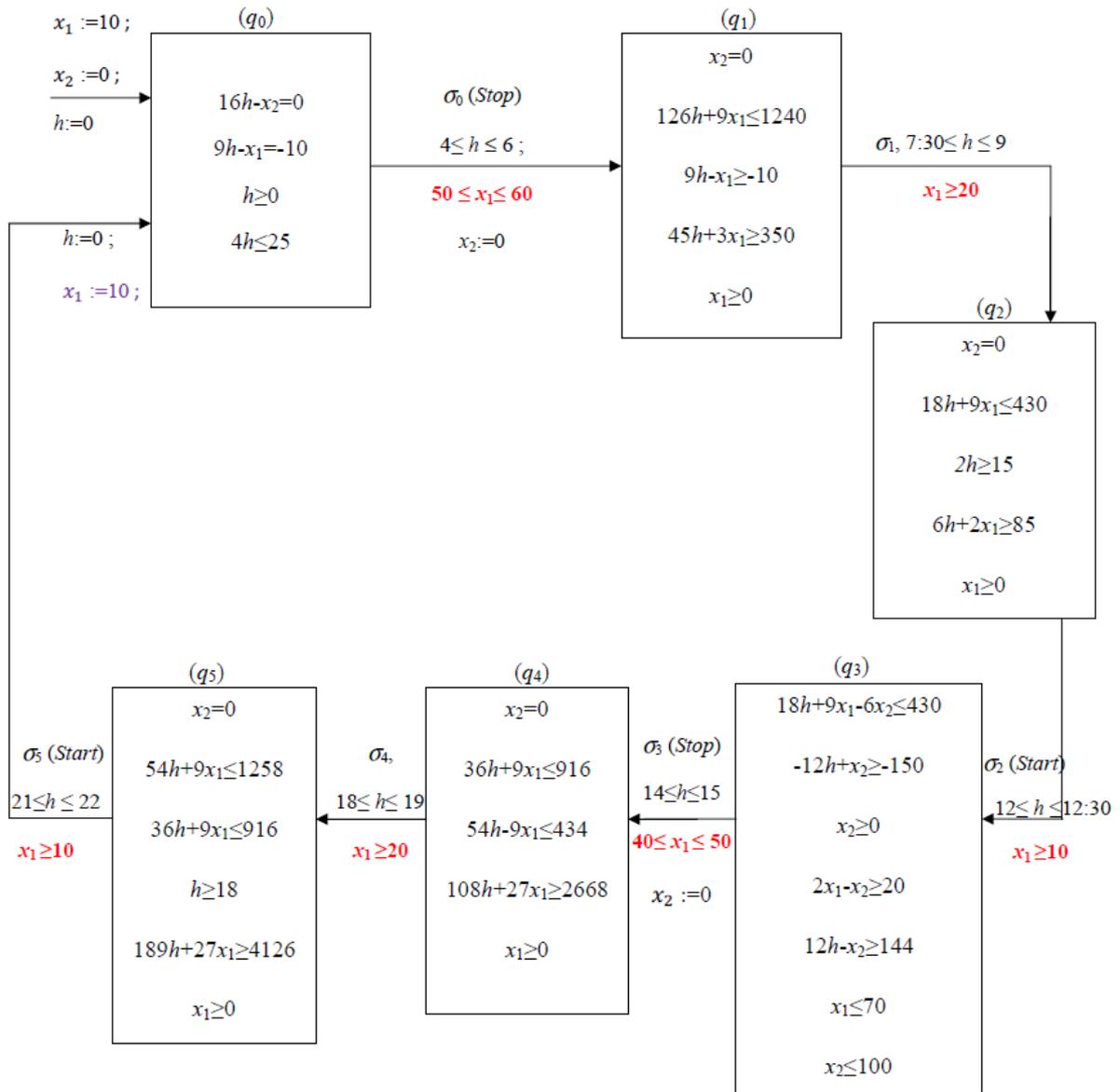


Fig.4.19- Automate constraintte atteignable du système d'approvisionnement en eau.

On a une seule boucle. On applique notre approche de commande maximale permissive donnée dans la section 4.3 en prenant en compte la nature de l'événement (proposition 4.3). On choisit comme point de départ le sommet  $(q_5)$ . On obtient les nouvelles gardes suivantes :

- $V_5 = \{21 \leq h \leq 21.63; x_1 = 10; x_2 = 0\} \Rightarrow$  Il y a réduction de l'intervalle de commande  $\Rightarrow$  l'événement  $\sigma_5$  étant contrôlable (arrêt de la pompe), la nouvelle garde est bien  $V_5$ .
- $V_4 = \{18 \leq h \leq 19; 22.8 \leq x_1 \leq 27.45; x_2 = 0\} \Rightarrow$  pas de réduction de l'intervalle de commande  $\Rightarrow$  l'événement  $\sigma_4$  étant incontrôlable, on poursuit le calcul.

- $V_3 = \{14.7 \leq h \leq 15; x_1=40; 32.4 \leq x_2 \leq 33.3\} \Rightarrow$  Il y a réduction de l'intervalle de commande  $\Rightarrow$  l'événement  $\sigma_3$  étant contrôlable (arrêt de la pompe), la nouvelle garde est bien  $V_3$ .
- $V_2 = \{12 \leq h \leq 12.5; x_1=10; x_2=0\} \Rightarrow$  Pas de réduction de l'intervalle de commande  $\Rightarrow$  l'événement  $\sigma_2$  étant contrôlable (démarrage de la pompe), la nouvelle garde est bien  $V_2$ .
- $V_1 = \{7.5 \leq h \leq 8.41; 22.7 \leq x_1 \leq 31.7; x_2=0\} \Rightarrow$  Réduction de l'intervalle de commande ( $h \leq 9 \rightarrow h \leq 8.41$ ) due à la contrainte avale ( $x_1 \geq 20$ )  $\Rightarrow$  l'événement  $\sigma_1$  étant incontrôlable  $\Rightarrow (V_1) \neq (U_1) \Rightarrow (q_2)$  devient interdit  $\Rightarrow$  **Il n'y a pas de solution au problème de commande.**

Un superviseur non vide peut être obtenu en modifiant les données du problème. Par exemple, pour que la commande n'ait pas à modifier la garde de la transition  $(q_1) \rightarrow (q_2)$ , une solution serait d'augmenter la production.

### 4.6 Conclusion

Dans ce chapitre, nous avons développé notre algorithme général de synthèse de commande maximale permissive des systèmes hybrides par automates hybrides rectangulaires.

Cet algorithme commence par la construction de l'automate contraint atteignable par analyse en avant du modèle automate du système auquel on aura ajouté les spécifications de commande désirées. L'automate contraint atteignable peut avoir une structure complexe, contenant des boucles et/ou des branches et pouvant avoir des sommets à plusieurs transitions de sortie. Notre procédure de commande doit alors tenir compte de toutes ces possibilités dans le modèle.

Ensuite, après avoir déterminé les nouvelles gardes de transition respectant les spécifications de commande que l'on souhaite imposer, nous avons construit le superviseur temporisé maximal permissif donnant les valeurs de l'horloge de commande qui autorise l'occurrence des événements contrôlables. Les intervalles obtenus correspondent à l'ensemble des trajectoires possibles du système contraint.

La terminaison de notre algorithme n'est pas garantie. Cependant, fort heureusement c'est généralement le cas pour les systèmes réels. Lorsqu'il termine, la commande calculée est maximale permissive.

Enfin, nous avons introduit quelques éléments de réponse au cas d'événements incontrôlables. Ce travail est encore en cours de traitement et représente une perspective intéressante à notre recherche.

# Conclusion générale et Perspectives

Le but de cette thèse est la synthèse d'une commande pour une classe de systèmes hybrides. Les démarches proposées dans ce domaine n'en sont qu'à leur début, ce qui nous a permis d'avoir un champ ouvert pour développer une approche originale de commande pour ce type de système.

C'est ainsi que nous nous sommes intéressés à l'élaboration d'une procédure de synthèse de commande pour les systèmes dynamiques hybrides, à forte composante événementielle, modélisés par un automate hybride. La mise en œuvre de cette approche est motivée par la capacité des modèles hybrides à représenter intrinsèquement les interactions entre la dynamique continue et la dynamique discrète d'un système réel. Ces modèles sont donc à la fois plus riches et plus complexes que les modèles des Systèmes à Événements Discrets temporisés qui ne représentent que l'information temporelle et peuvent par conséquent mener à une perte considérable d'informations nécessaires pour l'étude du comportement du système.

Dans notre cas, nous avons choisi de modéliser le système dynamique hybride (SDH) par un automate hybride rectangulaire vérifiant certaines hypothèses. Le choix de ce modèle nous a permis de contourner la complexité que pose l'analyse des modèles hybrides en général. Dans notre modèle, On définit les variables du système, leurs évolutions rectangulaires (dans un intervalle donné) ainsi que les contraintes physiques qui délimitent leurs séjours dans un état continu. On caractérise aussi les transitions discrètes d'un état à un autre suite à l'occurrence d'un événement contrôlable, auquel on associe un intervalle d'horloge. Lorsque le modèle du système est bien élaboré, on réalise une analyse d'atteignabilité en avant dans le but de déterminer les espaces d'état atteignables du système et d'étudier son comportement dynamique.

L'objectif de la commande consiste à associer des spécifications de commande au système se traduisant par des contraintes sur les gardes des transitions de l'automate. Ces dernières sont ajoutées indépendamment du comportement du processus, elles peuvent alors être inconsistantes avec la dynamique du système. On obtient l'automate contraint sur lequel on applique une analyse d'atteignabilité en avant pour calculer ses espaces d'état atteignables. Ces derniers sont réduits par rapport à ceux déterminés par l'analyse en avant de l'automate non contraint. Certains espaces peuvent aussi être supprimés, signifiant que les contraintes sont plus fortes que la dynamique du sommet correspondant.

Notre approche de commande se fait par la réalisation d'une procédure amont/aval pour la détermination de la nouvelle garde de transition entre deux sommets. Les calculs réalisés reposent sur la détermination de la durée de séjour, valeur contrainte par l'espace atteignable du sommet correspondant. L'originalité et en même temps la complexité de notre approche réside dans le fait qu'elle tient compte de la forte interaction amont/aval dans l'automate étudié.

Une fois la commande maximale permissive calculée pour une transition donnée, l'étape qui suit consiste à généraliser cette commande pour un automate et à construire le superviseur temporisé autorisant l'occurrence des événements contrôlables par les intervalles d'horloge. Ces intervalles sont déterminés par programmation linéaire et donne toutes les trajectoires possibles du système contraint. Lors de la considération de l'automate global, il nous faut tenir compte d'une structure générale. Pour cela, nous avons généralisé l'expression obtenue pour une transition simple au cas de sommets ayant plus d'une transition de sortie. L'expression de la commande maximale permissive pour un sommet se trouvant à l'amont d'un sommet « multi-sorties » fait intervenir la conjonction de toutes les nouvelles gardes avales préalablement calculées.

Lorsque l'algorithme général est établi, nous nous sommes proposé d'étendre cette procédure au cas d'événements incontrôlables. Celui-ci est encore en cours de traitement. Cependant, des premiers résultats ont été proposés.

Des cas d'études ont été traités afin d'appliquer notre approche de commande et de démontrer son originalité. Un autre cas nous a aussi permis de considérer le cas d'événements incontrôlables.

Plusieurs perspectives peuvent être envisagées, à court terme ou à long terme, comme suite au travail présenté dans cette thèse. Nous les avons regroupées selon trois axes :

Dans un premier axe, nous envisageons d'étudier la décidabilité de l'algorithme de commande maximale permissive développée dans ce travail. En effet, bien que l'algorithme termine pour les exemples traités dans ce mémoire, exprimant le fait qu'il y a une solution aux spécifications de commande désirées, sa terminaison n'est pas garantie dans le cas général. Par conséquent, on désire trouver les conditions nécessaires ou suffisantes de terminaison.

Dans un second axe, nous visons à implémenter la démarche de commande développée dans ce travail. Pour ce faire, nous utiliserons les différentes bibliothèques proposées dans la littérature, permettant la modélisation et l'analyse des automates hybrides rectangulaires. Une étude de la performance de l'algorithme proposé sera ensuite établie afin de permettre de choisir l'implémentation la plus optimale en termes de temps d'exécution de ressources consommées.

Dans un dernier axe, nous proposons de combiner notre approche de commande avec une approche de diagnostic hors-ligne que nous aurons préalablement développée. Ce travail peut être envisagé dans le cadre d'une extension de l'approche proposée dans (Sampath et al., 1998) pour les systèmes à événements discrets. Il s'agit de proposer une solution intégrée permettant de coupler le module de commande avec le module de diagnostic.



# Annexe A

## Synthèse de commande du système à deux tronçons de circulation

Nous présentons dans ce qui suit le développement détaillé de la commande du système à deux tronçons de circulation.

a) Point de départ : le transition  $(q_{14}) \rightarrow (q_{04})$  (figure A.1)

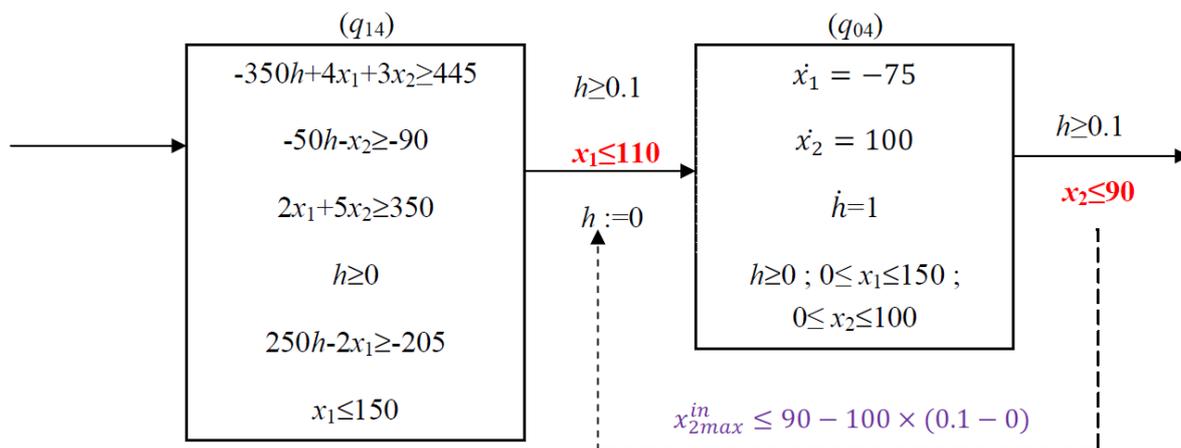


Fig.A.1- Calcul de la commande pour la transition  $(q_{14}) \rightarrow (q_{04})$ .

L'application du théorème 3.2 donne la nouvelle garde :

$$V_{14} = E_{14} \wedge \{h \geq 0.1; x_1 \leq 110\} \wedge \{x_2 \leq 80\}$$

Par programmation linéaire, on obtient :

$$V_{14} = \{0.1 \leq h \leq 0.53; 76.8 \leq x_1 \leq 110; 57.6 \leq x_2 \leq 63.5\}$$

b) Transition amont: le transition  $(q_{04}) \rightarrow (q_{14})$  (figure A.2)

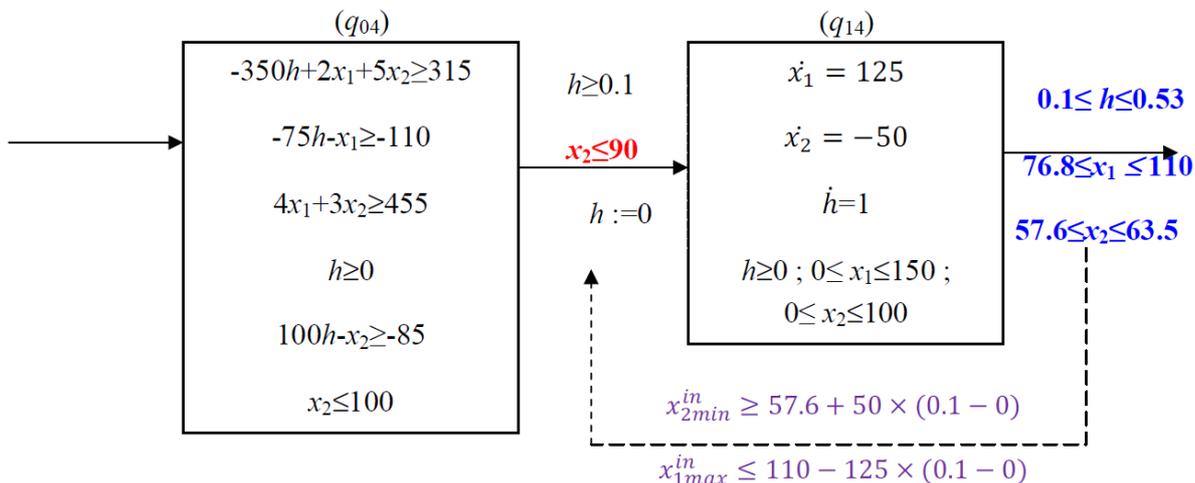


Fig.A.2- Calcul de la commande pour la transition  $(q_{04}) \rightarrow (q_{14})$ .

$$V_{04} = E_{04} \wedge \{h \geq 0.1; x_2 \leq 90\} \wedge \{x_2 \geq 62.6; x_1 \leq 97.5\}$$

$$\Rightarrow V_{04} = \{0.1 \leq h \leq 0.71; 56.75 \leq x_1 \leq 66.8; 62.6 \leq x_2 \leq 90\}$$

c) Transition amont: le transition  $(q_{14}) \rightarrow (q_{04})$  (figure A.3)

Il s'agit d'une vérification de la garde calculée  $V_{14}$  dans a) après le calcul de la nouvelle garde  $V_{04}$  dans b).

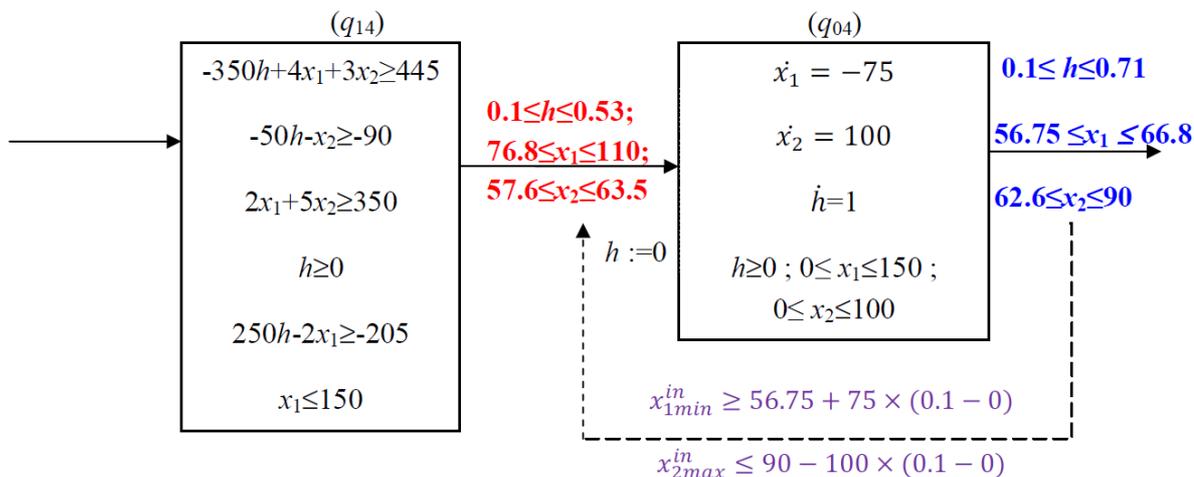


Fig.A.3- Calcul et vérification de la commande pour la transition  $(q_{14}) \rightarrow (q_{04})$ .

$V_{14}=E_{14}\wedge\{0.1\leq h\leq 0.53; 76.8\leq x_1\leq 110; 57.6\leq x_2\leq 63.5\}\wedge\{x_2\leq 80; x_1\geq 64.25\}\Rightarrow$  L'addition des contraintes aval ne change pas la garde  $V_{14}$  car elles sont plus faibles que les contraintes amont  $\Rightarrow V_{14}=\{0.1\leq h\leq 0.53; 76.8\leq x_1\leq 110; 57.6\leq x_2\leq 63.5\}$ .

- d) On sort de la boucle et on aborde la dernière transition de la branche :  $(q_{13})\rightarrow(q_{04})$  (figure A.4).

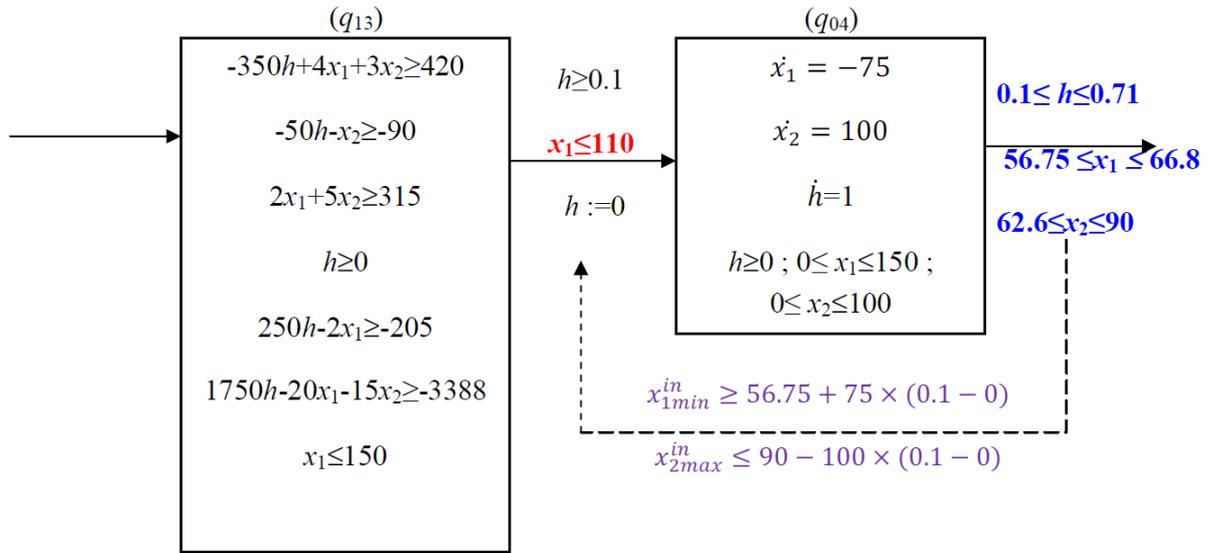


Fig.A.4- Calcul de la commande pour la transition  $(q_{13})\rightarrow(q_{04})$ .

$$V_{13}=E_{13}\wedge\{h\geq 0.1; x_1\leq 110\}\wedge\{x_2\leq 80; x_1\geq 64.25\}$$

$$\Rightarrow V_{13}=\{0.1\leq h\leq 0.58; 72.8\leq x_1\leq 110; 54.6\leq x_2\leq 61\}$$

- e) Le calcul des nouvelles gardes dans les transitions amont de la branche est identique à celui de la phase d). On obtient dans l'ordre de calcul:

- $(q_{03})\rightarrow(q_{13})$ :  $V_{03}=E_{03}\wedge\{h\geq 0.1; x_2\leq 90\}\wedge\{x_2\geq 59.6; x_1\leq 97.5\}\Rightarrow V_{03}=\{0.1\leq h\leq 0.78;$   
 $51.5\leq x_1\leq 59.6; 60.3\leq x_2\leq 90\}$
- $(q_{12})\rightarrow(q_{03})$ :  $V_{12}=E_{12}\wedge\{h\geq 0.1; x_1\leq 110\}\wedge\{x_2\leq 80; x_1\geq 59\}\Rightarrow V_{12}=\{0.1\leq h\leq 0.65;$   
 $67.2\leq x_1\leq 110; 50.4\leq x_2\leq 57.5\}$
- $(q_{02})\rightarrow(q_{12})$ :  $V_{02}=E_{02}\wedge\{h\geq 0.1; x_2\leq 90\}\wedge\{x_2\geq 55.4; x_1\leq 97.5\}\Rightarrow V_{02}=\{0.1\leq h\leq 0.85;$   
 $46.25\leq x_1\leq 54.7; 55.4\leq x_2\leq 90\}$
- $(q_{11})\rightarrow(q_{02})$ :  $V_{11}=E_{11}\wedge\{h\geq 0.1; x_1\leq 110\}\wedge\{x_2\leq 80; x_1\geq 53.75\}\Rightarrow V_{11}=\{0.1\leq h\leq 0.72;$   
 $61.6\leq x_1\leq 110; 46.2\leq x_2\leq 54\}$
- $(q_{01})\rightarrow(q_{11})$ :  $V_{01}=E_{01}\wedge\{h\geq 0.1; x_2\leq 90\}\wedge\{x_2\geq 51.2; x_1\leq 97.5\}\Rightarrow V_{01}=\{0.41\leq h\leq 0.8;$   
 $20\leq x_1\leq 49.1; 51.2\leq x_2\leq 90\}$

Finalement, les nouvelles gardes obtenues sont résumées dans la figure A.5.

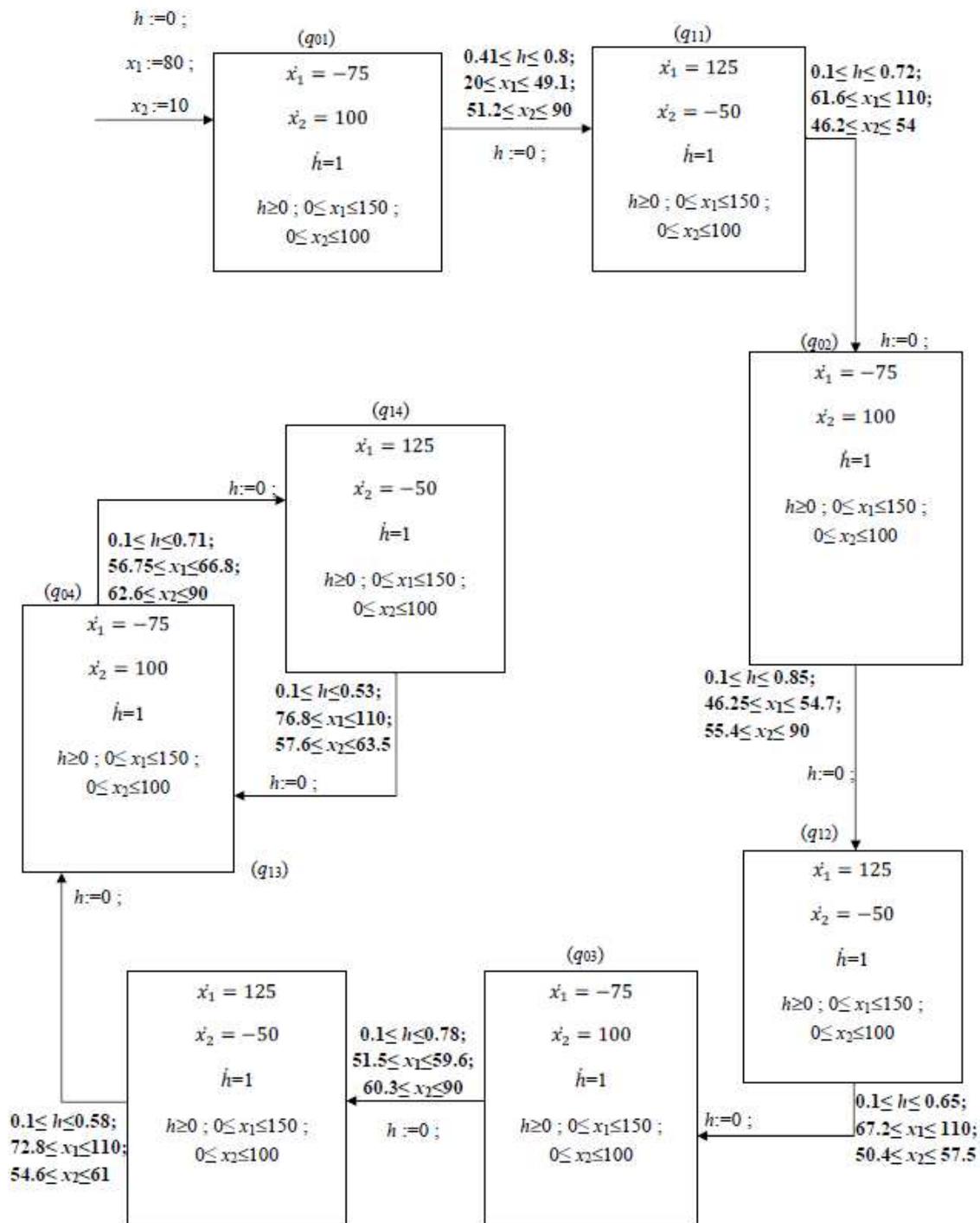


Fig.A.5- Nouvelles gardes pour le système à 2 tronçons de circulation.



## Annexe B

# Etude d'une deuxième configuration du système de production

Nous étudions dans ce qui suit une nouvelle configuration (figure B.1) du système de production introduit dans le chapitre 4. Celle-ci nous permettra de traiter le cas d'un système ayant un sommet « multi-sortie ».

On suppose que, à partir du sommet  $(q_1)$ , deux chemins sont possibles :

- 1- La consommation diminue dans le stock 1, d'où l'augmentation du flow  $x_1$ . Puis, la production change du stock 1 vers le stock 2:  $(q_1) \rightarrow (q_2) \rightarrow (q_3)$ .
- 2- La consommation augmente dans le stock 1, d'où la diminution du flow  $x_1$ . Puis, la production change du stock 1 vers le stock 2:  $(q_1) \rightarrow (q_2') \rightarrow (q_3)$ .

En effet, on suppose qu'il y a deux types de consommation à partir du sommet  $(q_1)$  qu'on aurait pu absorber en mettant un flux  $\dot{x}_1 = [2, 4]$  dans le sommet  $(q_2)$ , comme dans la première configuration, mais ce serait moins précis. On préfère alors diviser l'état en réduisant l'intervalle de consommation, à cause de l'incertitude sur cet élément.

La contrainte  $90 \leq x_1 \leq 100$  reste la même dans les deux chemins afin de garantir cette spécification lorsque la production change de stock.

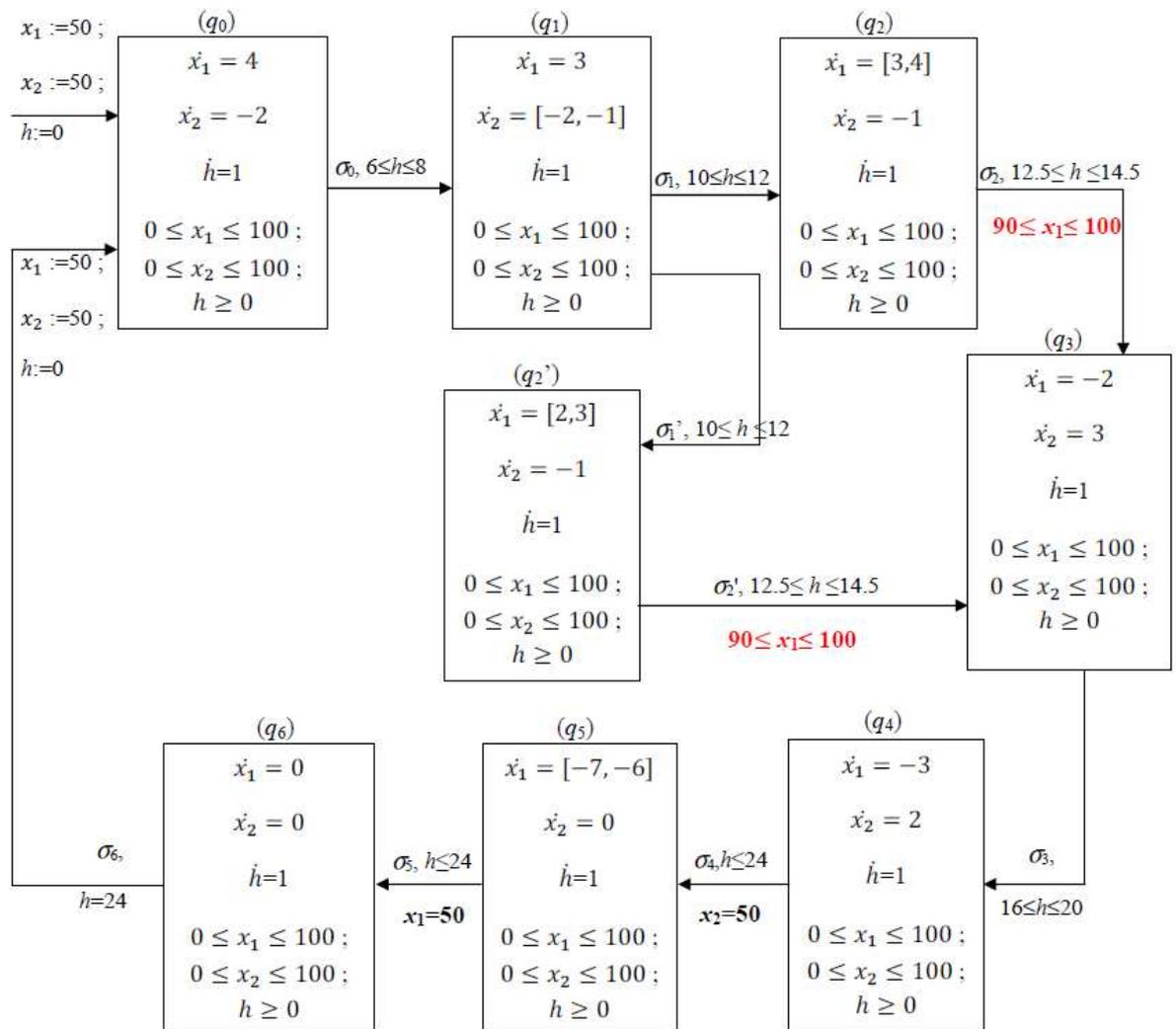


Fig.B.1- Nouvelle configuration du modèle automate du système de production.

L'analyse en avant de cet automate donne 12 sommets formant l'automate contraint atteignable de la figure B.2. On remarque que ce dernier est plus complexe que celui présenté dans le chapitre 4 (figure 4.14). En effet, on a deux boucles imbriquées et on se retrouve dans la situation où le sommet  $(q_1)$  possède deux transitions de sortie.

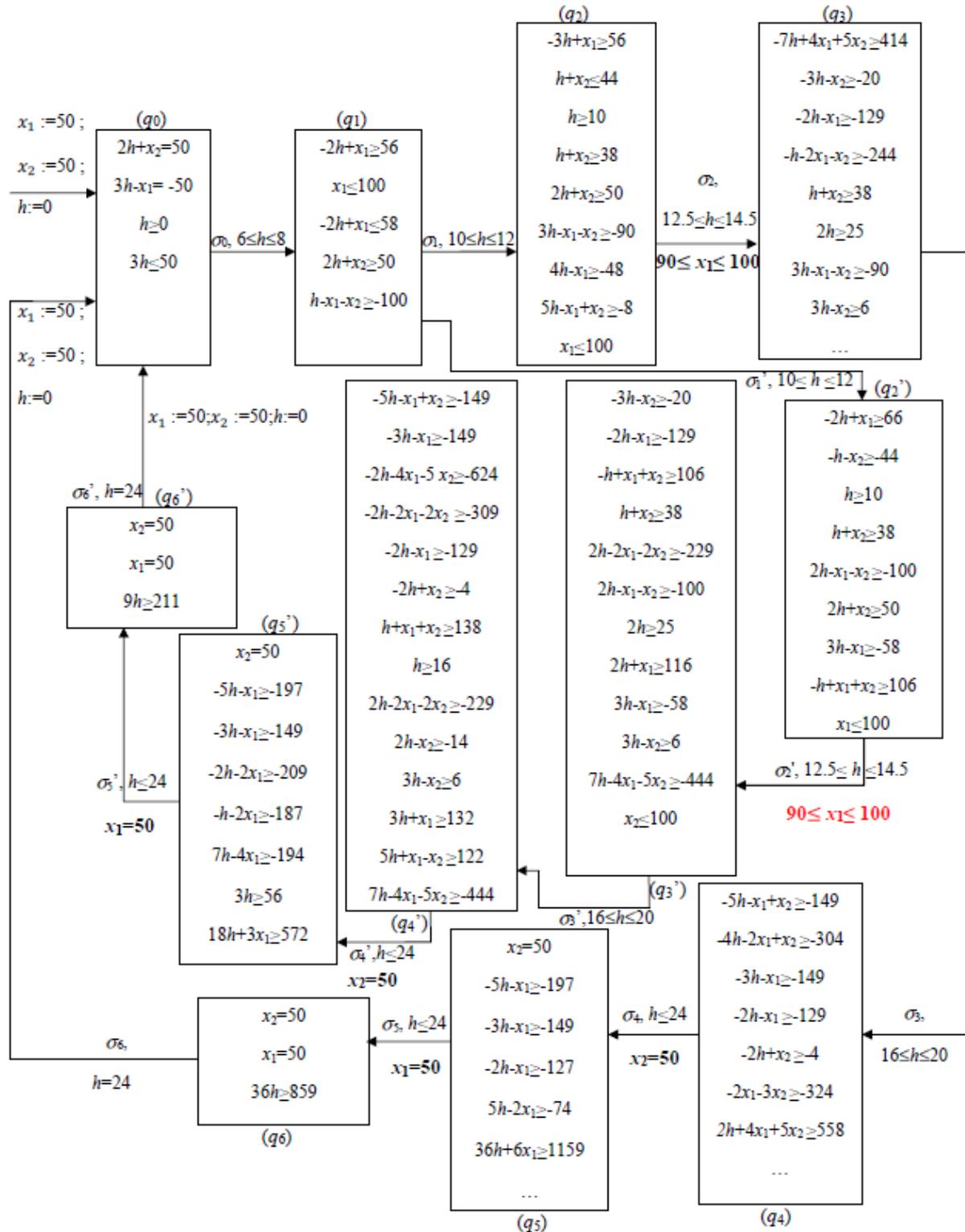


Fig.B.2- Nouvel automate contraint atteignable du système de production.

Dans ce cas, afin de déterminer le superviseur temporisé maximal permissif, on calcule tout d'abord les nouvelles gardes de la première boucle, allant de  $V_6$  en remontant jusqu'à  $V_1$  en appliquant le théorème 3.2, puis de la même manière, on calcule les nouvelles gardes de la

deuxième boucle, allant de  $V_6'$  en remontant jusqu'à  $V_1'$ . La nouvelle garde  $V_0$  en sortie du sommet  $(q_0)$  tiendra alors compte des deux nouvelles gardes  $V_1$  et  $V_1'$  en sortie du sommet  $(q_1)$ , comme indiqué par la proposition 4.1.

On obtient finalement le superviseur donné par la figure B.3.

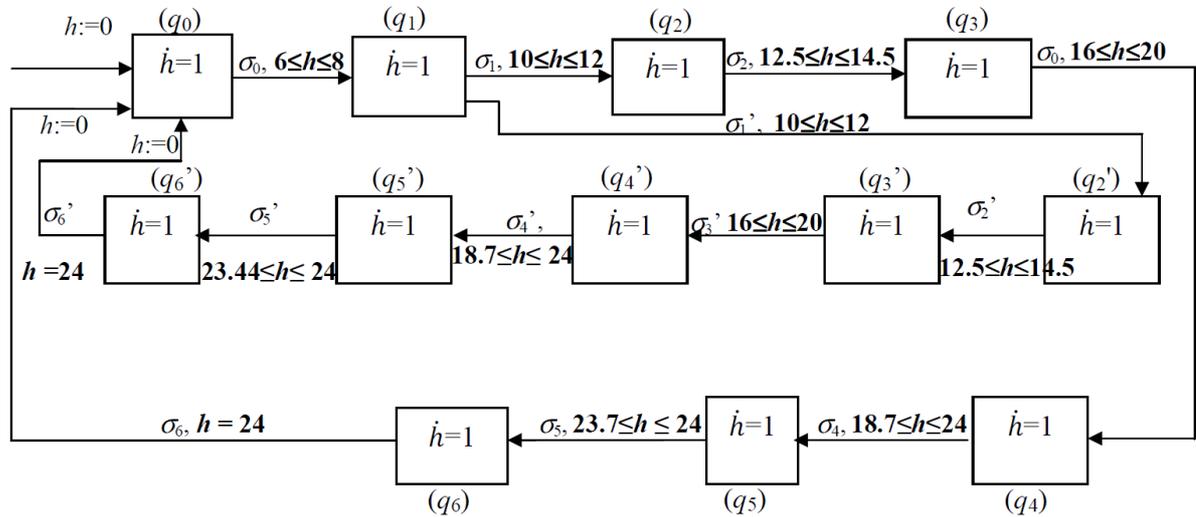


Fig.B.3- Nouveau superviseur temporisé maximal permissif du système de production.

# Bibliographie

- Alla, H. et R. David. 1998, « Continuous and hybrid petri nets », *Journal of Circuits, Systems and Computer*, vol. 8, n°1, p.159-188.
- Allam, M. et H. Alla. 1996, « From hybrid petri nets to hybrid automata », *Journal européen des systèmes automatisés*, vol. 320, n° 9-10, p. 1165-1185.
- Allam, M. 1998, *Sur l'analyse quantitative des RdP hybrides : une approche basée sur les automates hybrides*, thèse de doctorat, Institut National Polytechnique de Grenoble, Grenoble.
- Altisen, K., P. Bouyer, T. Cachat, F. Cassez et G. Gardey. 2005, « Introduction au contrôle des systèmes temps-réel », *European Journal of Automation*, vol. 39, n°1-2-3, p. 367-380.
- Altisen, K., G. Gößler, A. Pnueli, J. Sifakis, S. Tripakis et S. Yovine. 1999, « A framework for scheduler synthesis », *Proceedings of the 1999 IEEE Real-Time Systems Symposium*, Phoenix AZ USA, p. 154-163.
- Altisen, K., et S. Tripakis. 1999, « On-the-Fly Controller Synthesis for Discrete and Dense-Time Systems », *World Congress on Formal Methods (FM'99)*, vol. 1708 de *Lecture Notes in Computer Science*, Springer, p. 233-252.
- Altisen, K., et S. Tripakis. 2002, « Tools for Controller Synthesis of Timed Systems », *Proceedings of the 2<sup>nd</sup> Workshop on Real-Time Tools (RT-TOOLS'02)*, Proc. Published as Technical Report 2002-25, Uppsala University, Sweden.
- Alur, R. 1999, « Timed automata », *Theoretical Computer Science*, vol. 126, p. 183- 235.
- Alur, R., C. Courcoubetis, N. Halbwachs, T.A. Henzinger et P.-H. Ho, X. Nicollin, A. Olivero, J. Sifakis et S. Yovine. 1995, « The algorithmic analysis of hybrid systems », *Theoretical Computer Science*, vol. 138, p. 3-34.
- Alur, R., C. Courcoubetis, T.A. Henzinger et P.-H. Ho.1993, « Hybrid automata: An algorithmic approach to the specification and verification of hybrid systems », *Hybrid Systems, LNCS*, p.209-229.
- Alur, R. et D. L. Dill. 1994, « A theory of timed automata », *Theoretical Computer Science*, vol. 126, p. 183-235.
- Alur, R., T. Dang, F. Ivancic. 2003, « Progress on reachability analysis of hybrid systems using predicate abstraction », dans *Hybrid Systems: Computation and Control, LNCS*, vol. 2623, Springer-Verlag, Berlin, Heidelberg, p. 4-19.

- 
- Alur, R., T. A. Henzinger et H. Wong-Toi. 1997, « Symbolic Analysis of Hybrid Systems », dans *Proceedings of the 37th IEEE Conference on Decision and Control*, session invité, p. 1-6.
- Antsaklis, P. J, J.A. Stiver et M. Lemmon. 1993, « Hybrid system modeling and autonomous control systems », *Hybrid Systems, LNCS 736*, Springer-Verlag, p. 366-392.
- Asarin, E. et O. Maler. 1999, « As soon as possible: time control of timed automata », dans *Hybrid Systems: Computation and Control HSCC'99, LNCS*, vol. 1579, p. 19-30.
- Asarin, E., O. Maler, A. Pnueli et J. Sifakis. 1998, « Controller Synthesis for Timed Automata », dans *Proc. IFAC Symp. On System Structure & Control*, Elsevier Science, p. 469-474.
- Asarin, E., T. Dang, G. Frehse, A. Girard, C. L. Guernic et O. Maler. 2006, «Recent progress in continuous and hybrid reachability analysis», dans *Proceedings of the IEEE International Symposium on Computer-Aided Control System Design*, Technische Universität München, Munich, Allemagne.
- Badouel, E., P. Darondeau et L. Bernardinello. 1994, « Polynomial algorithms for the synthesis of bounded nets », dans *Proceedings Caap 95, Lecture Notes in Computer Science 915*, Springer, p. 364-378.
- Batis, S. et H. Alla. 2012a, « Timed control for rectangular hybrid systems », dans *Proceedings of the 14<sup>th</sup> IFAC Symposium on Information Control Problems in Manufacturing, INCOM'12*, Bucharest, Romania, p. 375-380.
- Batis, S. et H. Alla. 2012b, « Maximal permissive timed control for a class of hybrid systems », dans *Proceedings of the 11<sup>th</sup> Workshop on Discrete Event Systems, WODES'12*, Guadalajara, Mexico, p.157-162.
- Batis, S. et H. Alla. 2013a, «Control of a class of hybrid systems. Application to a water supply system », dans *Proceedings of the 10<sup>th</sup> IEEE International Conference on Networking, Sensing and Control, ICNSC'13*, Paris, France.
- Batis, S. et H. Alla. 2013b, «Controller Synthesis for a Class of Hybrid Systems », *12<sup>th</sup> biannual European Control Conference, ECC'13*, Zurich, Suisse.
- Batis, S. et H. Alla. 2013c, «Controller Synthesis for a Class of Hybrid Systems », soumis à *European Journal of Control (EJC)*.
- Bemporard, A., M. Morari. 1999, « Control of systems integrating logic, dynamics, and constraints », *Automatica*, vol. 35, n°3, p. 407-428.
- Bengtsson, J. et W. Yi. 2004, « Timed automata : Semantics, algorithms and tools », *Technical Report 316, UNU-IIST, Macau*.
- Berthomieu, B. et M. Diaz. 1991, « Modeling and verification of time dependent systems using time petri nets », *IEEE Transactions on Software Engineering*, vol. 17, n° 3, p. 259-273.

- 
- Brandin, B et W.M. Wonham. 1994, « Supervisory control of timed discrete event systems », *IEEE Transactions on Automatic Control*, vol. 39, n°2, p. 329- 341.
- Branicky, M.S. 1998, « Multiple Lyapunov functions and other analysis tools for switched and hybrid systems », dans *IEEE Transaction on Automatic Control, Special Issue on Hybrid Systems*, vol. 43, p. 475-482.
- Branicky, M.S., V.S. Borkar et S. K. Mitter. 1994, « A unified framework for hybrid control », dans *Proceedings of the 33<sup>rd</sup> Conference on Decision and Control*, p. 4228-4234.
- Branicky, M.S. et S. K. Mitter. 1995, « Algorithms for optimal hybrid control », dans *Proceedings of the 34<sup>th</sup> Conference on Decision and Control*, p. 2661-2666.
- Brockett, R. 1993, « Hybrid models for motion control systems », *Essays in Control: Perspectives in the Theory and its Applications*, p. 29-53.
- Brockett, R. 1994, « Language drive hybrid systems », dans *Proceedings of the 33<sup>rd</sup> Conference on Decision and Control*, p. 4210-4214.
- Cassandras, C. et S. Lafortune. 2008, *Introduction to Discrete Event Systems*, Second Edition, Springer.
- Cassez, F. et N. Markey. 2007, « Contrôle et Implémentation des systèmes temporisés », *Actes de l'école d'été ETR'07*.
- Cassez, F. et O. H. Roux. 2006, « Structural translation from time petri nets to timed automata », *Journal of Software and Systems*, vol. 79, p. 1456-1468.
- Cassez, F., A. David, E. Fleury, K.G. Larsen et D. Lime. 2005, « Efficient On-The-Fly Algorithms for the Analysis of Timed Games », *Proceedings of the 16<sup>th</sup> International Conference on Concurrency Theory (CONCUR'2005)*, Lecture Note in Computer Science, Springer, vol. 3653, p. 66-80.
- Chen, Y. et G. Provan. 1997, « Modeling and diagnosis of timed discrete event systems- a factory automation example », dans *The American Control Conference, New Mexico*, p. 31-36.
- Dang, T. 2000, *Vérification et synthèse des systèmes hybrides*, thèse de doctorat, Institut National Polytechnique de Grenoble, France.
- David, R. et H. Alla.. 1989, *Du Grafctet aux réseaux de Petri*, Hermes Science Publications, Paris.
- David, R. et H. Alla. 2010, *Discrete, Continuous and Hybrid Petri Nets*, Springer, Berlin Heidelberg.
- De Alfaro, L., T.A. Henzinger et R. Majumdar. 2001, « Symbolic Algorithms for Infinite-State Games », *Proceedings of the 12<sup>th</sup> International Conference on Concurrency Theory (CONCUR'01)*, vol. 2154 de *Lecture Notes in Computer Science*, Springer, p. 536- 550.
- De Alfaro, L., M. Faëlla, T.A. Henzinger, R. Majumdar et M. Stoelinga. 2003, «The Element of Surprise in Timed Games », *Proceedings of the 14<sup>th</sup> International Conference on Concurrency Theory (CONCUR'01)*, vol. 2761 de *Lecture Notes in Computer Science*, Springer, p. 142- 156.
-

- 
- Derbel, H. 2009, *Diagnostic à base de modèles des systèmes temporisés et d'une sous-classe des systèmes dynamiques hybrides*, thèse de doctorat, L'université Joseph Fourier, Grenoble.
- Faëlla, M., S. La Torre et A. Murano. 2002, « Dense Real-Time Games », *Proceedings of the 17th IEEE Symposium on Logic in Computer Science (LICS'02)*, IEEE Computer Society Press, o. 167-176.
- Frehse, G. 2005, « Phaver: Algorithmic verification of hybrid systems past hytech », dans *Proceedings of the 5<sup>th</sup> International Workshop on Hybrid Systems: Computation and Control*, p. 258-273.
- Gaubert, S. 1995, « Performance evaluation of (max, +) automata », *IEEE Transactions on Automatic Control*, vol. 40, n°12, p. 2014-2025.
- Ghaffari, A., N. Rezg et X. Xie. 2003, « Design of a live and maximally permissive Petri net controller using the theory of regions », *IEEE Transactions on Robotics and Automation*, vol. 19, n°1, p. 137-141.
- Gilbert, J. Ch. 2007, *Ensembles convexes, Éléments d'optimisation différentiable : Théorie et Algorithmes*, p. 13-45, sera publié.
- Giua, A., F. Dicesare et M. Silva. 1992, « Generalized mutual exclusion constraints on nets with uncontrollable transitions », dans *Proceedings IEEE International Conference on Systems, Manufacturing and Cybernetics*, vol. 2, Chicago (IL), p. 974-279.
- Giua, A. et E. Usai. 1996, « High-level petri nets : A definition », dans *Proceedings of the 35<sup>th</sup> Conference on Decision and Control*, Kobe, Japon, p. 148-150.
- Guéguen, H. et J. Zaytoon. 2004, « On the formal verification of hybrid systems », *Control Engineering Practice*, vol. 12, n° 10, p. 1253-1267.
- Harel, D. 1987, « Statecharts : A visual formalism for complex systems », *Sci. Comput. Programming* 8, p. 231-274.
- Harel, D., A. Pnueli, J. Schmidt et R. Sherman. 1987 « On the Formal Semantics of Statecharts », dans *Proc. 2nd IEEE Symp. on Logic in Computer Science, Ithaca, NY*, p. 54-64.
- Henzinger, T.A., B. Horowitz, R. Majumdar. 1999, « Rectangular hybrid games », dans *Proceedings of the Tenth International Conference on Concurrency Theory, Lecture Notes in Computer Science 1664*, Springer-Verlag, p. 320-335.
- Henzinger, T. et P. Kopke. 1999, « Discrete-time control for rectangular hybrid automata », *Theoretical Computer Science*, vol. 221, p. 369-392.
- Henzinger, T. et P. Kopke. 1996, « State Equivalences for Rectangular Hybrid Automata », dans *Proceedings of the Seventh International Conference on Concurrency Theory (CONCUR)*, Lecture Notes in Computer Science 1119, Springer, p. 530-545.
- Henzinger, T., P. Kopke, A. Puri et P. Varaiya. 1998, « The what's decidable about hybrid automata? », *Journal of Computer and System Sciences*, vol. 57, p. 94-124.
-

- Henzinger, T. A. 1996, « The theory of hybrid automata », *Hybrid Systems II, LNCS*, vol. 999, p. 278-292.
- Henzinger, T. A., P. H. Ho, H. W. Toi. 1997, « Hytech : A model checker for hybrid systems », *International Journal on Software Tools for Technology Transfer*, vol. 1, n°1-2, p. 110-122.
- Henzinger, T. A. et R. Majumdar. 2000, « Symbolic model checking for rectangular hybrid systems », dans *TACAS 2000: Tools and algorithms for the construction and analysis of systems, Lecture Notes in Computer Science*, New-York, Springer-Verlag, p. 142-156.
- Henzinger, T. A. et V. Rusu. 1998, « Reachability Verification for Hybrid Automata », dans *Proceedings of the First International Workshop on Hybrid Systems: Computation and Control (HSCC)*, Lecture Notes in Computer Science 1386, Springer, p. 190-204.
- Holloway, L. et B. Krogh. 1990, « Synthesis of feedback control logic for a class of controlled Petri nets », *IEEE Transactions on Automatic Control*, vol. 35, n°5, p. 514-523.
- Hopcroft, J., R. Motwani et J. Ullman. 2007, *Introduction to Automata Theory, Languages and Computation*, 3<sup>rd</sup> Edition, Prentice Hall.
- Iodarche, M., J. Moody et P. Antsaklis. 2001, « A method for the synthesis of liveness enforcing supervisors in Petri nets », dans *Proceedings IEEE American Control Conference (ACC 2001)*, vol. 6, Arlington (VA), p. 4943-4948.
- Kopke, P. W. 1996, *The Theory of Rectangular Hybrid Automata*, thèse de doctorat, Cornell University, NY, USA.
- Koutsoukos, X.D, P.J. Antsaklis, J. A. Stiver et M.D. Lemmon. 2000, « Supervisory Control of Hybrid Systems », *Proceedings of The IEEE*, vol. 88, n° 7, p. 1026-1049.
- Kurovsky, M. 2002, *Etude des systèmes dynamiques hybrides par représentation d'état discrète et automate hybride*, thèse de doctorat, L'université Joseph Fourier, Grenoble.
- Le Bail, J., H. Alla, R. David. 1991, « Hybrid Petri Nets », dans *Proceedings of the European Control Conference*, Grenoble- France, p. 1472- 1477.
- Lygeros, J., D.N. Godbole et S. Sastry. 1996, « Multiagent hybrid system design using game theory and optimal control », dans *Proceedings of the 35<sup>th</sup> IEEE Conference on Decision and Control, Kobe, Japon*, p. 1190- 1195.
- Lygeros, J., D.N. Godbole et S. Sastry. 1998, « Verified hybrid controllers for automated vehicles », dans *IEEE Transactions on Automatic Control, Special Issue on Hybrid Systems*, vol. 43, p. 522- 539.
- Lygeros, J., C. Tomlin et S. Sastry. 1999, « Controllers for reachability specifications for hybrid systems », *Automatica*, vol. 35, p. 349- 370.
- Maler, O., A. Pnueli et J. Sifakis. 1995, « On the Synthesis of Discrete Controllers for Timed Systems », dans *Proc. 12<sup>th</sup> Symp. On Theoretical Aspects of Computer Science (STACS'95)*, Springer, vol. 900, p. 229-242.
- Manon, P. et C. Valentin- Roubinet. 1999a, « Synthesis of a trajectory to nominal mode for a class of hybrid dynamical systems », *JESA*, vol. 33, p. 995-1014.

- 
- Manon, P. et C. Valentin- Roubinet. 1999b, « Controller synthesis for hybrid systems with linear vector fields », *Proceedings of the workshop ISIC*, Cambridge, USA.
- Mendler, M. et G. Lüttgen. 2001, « Statecharts: from visual syntax to model-theoretic semantics », dans *Workshop on Integrating Diagrammatic and Formal Specification Techniques (IDFST 2001)*, p. 615–621.
- Merlin, P. M. 1974, *A study of the recoverability of computing systems*, thèse de doctorat, Department of Information and Computer Science, University of California.
- Moody, J. et P. Antsaklis. 1998, *Supervisory Control of Discrete Event Systems Using Petri Nets*, Kluwer Academic Publishers, Boston.
- Moody, J. et P. Antsaklis. 2000, « Petri net supervisors for DES with uncontrollable and unobservable transitions », *IEEE Transactions on Automatic Control*, vol. 45, n°3, p. 462-476.
- Mosterman, P. J. 1997, *Hybrid Dynamic Systems: a Hybrid Bond Graph Modeling Paradigm and its Application in Diagnosis*, thèse de doctorat, Vanderbilt University.
- Müller, O. et T. Stauner. 2000, « Modelling and verification using linear hybrid automata- a case study », *Mathematical and Computer Modelling of Dynamical Systems*, vol. 6, n°1, p.71-89.
- Murata, T. 1989, « Petri nets : Properties, analysis and applications », dans *Proceedings of the IEEE*, vol. 77, n°4, p. 541-580.
- Olivero, A. 1994, *Modélisation et analyse des systèmes temporisés et hybrides*, Thèse de doctorat, Institut National Polytechnique de Grenoble.
- Petri, C. A. 1962, *Kommunikation mit automaten*, thèse de doctorat, University of Bonn.
- Petersson, S. et B. Lennartson. 1995, « Hybrid modeling focused on hybrid petri nets », dans *Proceedings of the 2<sup>nd</sup> European Workshop on Real-Time and Hybrid Systems*, Grenoble, France, p. 303-309.
- Petersson, S. et B. Lennartson. 1996, « Time-optimal control and disturbance compensation for a class of hybrid systems », *Proceedings of IFAC'96*, San Fransisco.
- Philippot, A. 2006, *Contribution au diagnostic décentralisé des systèmes à événements discrets : Application aux systèmes manufacturiers*, thèse de doctorat, Université de Reims Champagne Ardenne.
- Pnueli, A. et R. Rosner. 1989, « On the Synthesis of a Reactive Module », *Proceedings of the 16<sup>th</sup> ACM Symposium on Principles of Programming Languages (POPL'89)*, ACM, p. 179-190.
- Puri, A., V. Borkar et P. Varaiya. 1996, «  $\mathcal{E}$ -Approximation of Differential Inclusions », Dans *Proceedings of Hybrid Systems III Workshop: Verification and Control*, vol. 1066 of *Lecture Notes in Computer Science*, p. 362-376.
- Ramadge, P. et W. Wonham. 1987a, « On the supremal controllable sublanguage of a given language », *SIAM J. Control Optim*, vol. 25, n°3, p. 637-259.
-

- 
- Ramadge, P. et W. Wonham. 1987b, «Supervisory control of a class of discrete event processes », *SIAM J. Control Optim*, vol. 25, n°1, p. 206-230.
- Ramadge, P. et W. Wonham. 1989, «The Control of Discrete Event Systems», dans *Proceedings IEEE: Special Issue on Dynamics of Discrete Event Systems*, vol. 77, n°1, p. 81-98.
- Roux, O. et V. Rusu. 1996, «Uniformity for the decidability of hybrid automata », dans *Proceedings of the 8th Conference on Computer Aided Veri CAV'96*, volume 1145 of *LNCS*, p. 301-316.
- Sampath, M., S. Lafortune et D. Teneketzis. 1998, « Active diagnosis of discrete-event systems », *IEEE Transactions on Automatic Control*, vol. 43, n°7, p. 908-929.
- Sava, A. T et H. Alla. 2001, « Commande par supervision des systèmes à événements discrets temporisés », *Modélisation des systèmes réactifs, MSR'01*, p. 71-86.
- Sava, A. 2002, *Sur la synthèse de la commande des systèmes à événements discrets temporisés*, thèse de doctorat, Département d'Automatique de Gipsa-lab, Institut Nationale Polytechnique de Grenoble.
- Spathopoulos, M. 2000, « Supervisory Control for Rectangular Hybrid Automata », dans *Proceedings of the 39<sup>th</sup> IEEE Conference on Decision and Control*, p. 35-41.
- Sreenivas, R. et B. Krogh. 1992, « On Petri nets models of infinite state supervisors », *IEEE transactions on Automatic Control*, vol. 37, n°2, p. 274-277.
- Stiver, J. A et P.J. Antsaklis. 1993, « On the controllability of hybrid control systems », *Proceedings of the 32<sup>nd</sup> IEEE CDC*, San Antonia, Etats-Unis.
- Tavernini, L. 1987, « Differential automata and their discrete simulators », *Non-linear Analysis, Theory, Methods and Application*, vol. 11, n°6, p. 6658-683.
- Thomas, W. 1995, « On the Synthesis of Strategies in Infinite Games », *Proceedings of the 12<sup>th</sup> Annual Symposium on Theoretical Aspects of Computer Science (STACS'95)*, vol. 900, Springer, p.1-13, Invited Talk.
- Tittus, M. 1995, *Control synthesis for batch processes*, Rapport technique n° 280, Control Engineering Lab, Chalmers university of technology, Gothenburg, Suède.
- Tittus, M. et B. Edgart. 1998, « Control design for integrator hybrid systems », *IEEE Transactions on automatic Control, Special Issue on Hybrid Systems*, vol. 43, p. 491-500.
- Tomlin, C. 1998, *Hybrid Control of Air Traffic Management Systems*, thèse de doctorat, Université de Californie, Berkeley.
- Tomlin, C., G.J. Pappas et S. Sastry. 1998, « Conflict resolution for air traffic management : A study in multi-agent hybrid systems », dans *IEEE Transactions on Automatic Control, Special Issue on Hybrid Systems*, vol. 43, p. 509-521.
- Uzam, M. et W. Wonham. 2006, « A hybrid approach to supervisory control of discrete event systems coupling RW supervisors to Petri nets », *International Journal of Advanced Manufacturing Technology*, vol. 28, n°7-8, p. 747-760.

- Vasiliu, A.I. 2012, *Synthèse de contrôleurs des systèmes à événements discrets basée sur les réseaux de Petri*, Thèse de doctorat, Université de Grenoble.
- Witsenhausen, H. S. 1966, « A class of hybrid-state continuous-time dynamic systems », *IEEE Transactions in Automatic Control*, vol.11, n°2, p. 161-167.
- Wong-Toi, H. 1997, « The Synthesis of Controllers for Linear Hybrid Automata », *Proceedings of the 36<sup>th</sup> IEEE Conference on Decision and Control*, IEEE Computer Society Press, p. 4607-4612.
- Wu, L. et M. Zhou. 2004, « Elementary siphons of Petri nets and their application to deadlock prevention in flexible manufacturing systems », *IEEE Transactions on Systems, Manufacturing and Cybernetics*, vol. 34, n°1, p. 38-51.
- Yamalidou, K., J. Moody, M. Lemmon et P. Anstaklis. 1996, « Feedback control of Petri nets based on place invariants », *Automatica*, vol. 32, n° 1, p. 15-28.
- Yovine, S. 1993, *Méthodes et outils pour la vérification symbolique des systèmes temporisés*, Thèse de doctorat, Institut National Polytechnique de Grenoble.
- Yovine, S. 1998, « Model checking timed automata », dans *Lectures on Embedded Systems*, G. Rozenberg and F. Vaandrager (Eds.). *Lecture Notes in Computer Science 1494*, Springer-Verlag, p. 114-152.
- Zaytoon, J. 2001, *Systèmes dynamiques hybrides*, Hermès.

