

République Algérienne Démocratique et Populaire
Ministère de l'enseignement Supérieur et de la Recherche
Scientifique



Ecole Nationale Polytechnique
Département de Génie Électrique
Spécialité Automatique

Projet de fin d'étude

**S
U
J
E
T**

Etude des Systèmes dynamiques hybrides:

Commande et observation

Dirigé par :

Mr M.Tadjine

Mr M.S.Bouchrit

Effectué par :

Mlle Cherif Lallia

Mlle Herizi Karima

Promotion: 2005-2006

Remerciements

On tient tout d'abord à remercier monsieur M.Tadjine et monsieur M.S.Boucherit d'avoir proposé et dirigé ce sujet d'actualité, qui nous a permis d'être à la pointe de l'innovation dans l'automatique.

On remercie monsieur F.Boudjema et Dj.Boukhetala pour avoir accepté d'être le président et l'examineur du jury. L'attention et l'intérêt qu'ils ont portés à notre travail nous honore .

Nos remerciements vont également à monsieur Kh.Benmensour pour son aide.

Dédicaces

Je dédie ce travail :

A mes très chers parents qui m'ont soutenue ;

A mes frères et sœurs et leurs enfants ;

A ma famille ;

A tous mes amis .

Karima

Dédicaces

Je dédie ce travail :

A mes très chers parents qui m'ont soutenue ;

A mes frères et sœurs ;

A ma famille ;

A tous mes amis .

Lallia

ملخص

هذا العمل يهدف إلى دراسة الأنظمة الهجينة. تطرقنا إلى تعريف هذه الأنظمة وإعطاء مختلف طرق تمثيلها ثم دراسة خاصية الاستقرار بعرض أهم النتائج. وبعدها دراسة الروايد الخاصة بهذا النوع من الأنظمة وتم تطبيق الراسد ذو نموذج الإنزلاق على مثال. وقد إعتدنا على الأمثلة لتوضيح كل هذه النقاط وأخيرا تم عرض تطبيقين الأول مكون من حوضين والهدف هو التحكم في منسوب الماء في الحوض والثاني مطبّر مخفض بغرض التحكم في توتر الخروج بالنسبة لمحاكاة هذه الأنظمة تتم بوسائل خاصة.

كلمات المفتاح: نظام هجين, التحكم, الراسد.

Résumé

Ce projet de fin d'études dont le but est l'études des systèmes dynamiques hybrides a débuté par des généralités et des définitions de ces systèmes ,en donnant les différents moyens de modélisation et de représentation , suivi d'une étude de la stabilité ou a été résumé les principaux résultats obtenus dans ce domaine , puis des observateurs à modes glissants ont été adapté au problème des systèmes hybrides , enfin une approche accès sur la commande optimale avec et sans contraintes a été présenté ,afin d'illustrer ces différents points et de montrer l'efficacité de la modélisation hybride ,deux applications ,l'une hydraulique sur la régulation du niveau d'eau de deux réservoirs et l'autre électrique sur la régulation de tension d'un convertisseur abaisseur DC-DC , ont été donnés sans oublier les différents exemples qui ont été illustrés tout au long des chapitres . La simulation des différents exposés nécessitant des moyens autres que la programmation Matlab a été effectuée en utilisant stateflow et la boîte à outils multi paramétrique spécifique aux systèmes hybrides récemment mise au point .

Mots clés : Système hybride, commande, observateur.

Abstract

This project of end of studies of which the goal is the studies of the hybrid dynamic systems begin with general information and definitions from these systems,en giving the various means from modeling and representation, follow-up of a study of stability or was summarized the principal results obtained in this field, then observers with sliding modes were adapted to the problem of the hybrid systems, finally an approach access on the optimal control with and without constraints was presented, in order to illustrate these various points and to show the effectiveness of hybrid modeling, two applications, one hydraulic on the regulation of the water level of two tanks and the other electric on the regulation of tension of a Buck converter DC-DC, were given ,without forgetting the various examples which were illustrated throughout the chapters. The simulation of the various systems requiring of the means other than Matlab was carried out by using the stateflow and with multi parametric toolbox specific to the hybrid systems recently developed.

Key words : Hybrid system ,control ,observer .

Sommaire

Introduction générale	1
Chapitre 1 : Généralités sur les systèmes dynamiques hybrides	3
1.1 Définition des systèmes dynamiques hybrides	3
1.1.1 Interactions entre les systèmes : phénomènes hybrides	3
1.1.2 Définition formelle des SDH et classification des phénomènes hybrides	5
1.2 Représentations des systèmes dynamiques hybrides	7
1.2.1 Modèle mixte dynamique et logique	7
1.2.2 Modèle affine par morceaux	8
1.3 Outils de simulation	9
1.4 Exemples de systèmes hybrides	10
1.4.1 La balle rebondissante	10
1.4.2 Thermostat	12
1.4.3 Le Convertisseur Boost	14
Chapitre 2 : Etude des systèmes dynamiques hybrides	18
1. La stabilité	18
1.1. Résultats généraux sur la stabilité des S.D.H	19
1.2. Stabilité et stabilisation des systèmes hybride linéaires à commutation de modèle	20
2. Observateurs pour les systèmes dynamiques hybrides	25
2.1 Position du Problème	25
2.2. L'interaction entre l'observateur continu et discret	26
2.3. Observateur par modes glissants	27
2.3.1 Observateur par mode glissant d'ordre 1	27
3. Commande optimale des systèmes hybrides	30
3.1. Formulation du problème de commande optimale	30
3.2. La commande optimale sans contraintes des systèmes commutés	30
3.3. Commande optimale avec contrainte	31
3.4. Détermination en temps réel de l'entrée de commande	32
3.5. Détermination off line de loi de commande du retour d'état	33
3.6. Receding Horizon Control	33
Chapitre 3: Applications	34
Application A : Système à deux réservoirs	34
Application B : Le convertisseur Buck DC-DC	42
Conclusion générale et perspective	57
Annexes	

Introduction Générale :

L'appellation « systèmes dynamiques hybrides » désigne des catégories de systèmes faisant intervenir explicitement et simultanément des phénomènes ou des modèles de type dynamique continu et de type dynamique événementiel. Ces systèmes sont classiquement constitués de processus continus interagissant avec ou supervisés par des processus discrets [1]. Ils résultent également de l'organisation hiérarchique de systèmes de contrôle/commande complexes ou de l'interaction entre des algorithmes discrets de planification et des algorithmes continus de commande.

Une théorie générale et unifiée pour modéliser, analyser, concevoir des commandes pour de tels systèmes n'est pas disponible actuellement [2]. Cependant, beaucoup de chercheurs travaillent sur des sous-classes de systèmes hybrides pour lesquelles de telles approches pourraient être développées. Différents formalismes sont proposés pour modéliser les systèmes hybrides. Les plus connus sont les automates hybrides [3], réseaux de Pétri hybrides [4], la structure modulaire [5], les systèmes affines par morceaux (PWA)[6] et les systèmes mixtes dynamiques et logiques (MLD)[7].

La nature même des systèmes hybrides rend leur étude, aussi bien théorique que pratique, plus délicate que celle des systèmes continus. Le caractère hybride des systèmes génère également des difficultés dans la résolution de problèmes.

Les approches de modélisation en automatique sont basés sur :

- des modèles d'équations différentielles et aux différences (états continu, à temps continu ou discret), modèles fréquentiels pour les systèmes continus [8],
- des modèles états-transitions et modèles markoviens [9] pour les systèmes événementiels.

Les méthodes d'analyse « classiques » prennent en compte un seul aspect à la fois, l'aspect continu ou l'aspect événementiel.

La plupart des systèmes réels sont composés de sous-processus continus (moteurs, procédés chimiques, systèmes de freinage) qui sont démarrés, reconfigurés et arrêtés par une commande logique, à état discrets (ordinateur, automate programmable). L'évolution d'un système réel est à la fois continue et événementielle. Pour garantir le bon fonctionnement d'un ensemble automatisé réel il est nécessaire de prendre en compte simultanément les aspects continus et événementiels de sa dynamique[10]. Les systèmes dynamiques hybrides (SDH) ont été introduits pour répondre à cette demande.

De nombreux problèmes mal traités par les approches homogènes sont résolus par l'approche hybride:

- variation de structure liée aux différents modes de marche [11],
- variation de paramètre lors de régimes de fonctionnement en grandes transitions [12],
- commande discontinue (par exemple par relais ou impulsions) [13],
- modélisation de phénomènes transitoires rapides par des commutations de modèles[14] .

La dernière décennie a vu croître l'intérêt porté à la recherche concernant les systèmes hybrides car ils couvrent plusieurs domaines d'application. Les progrès rapides de l'informatique et de la technologie des ordinateurs encouragent les analyses des systèmes hybrides car les ordinateurs sont capables de résoudre des problèmes de calcul de plus en plus complexes [15].

Ainsi, on utilise le formalisme des Réseaux de Petri hybrides ou de la structure modulaire en utilisant les réseaux de Petri pour la partie discrète pour la modélisation et la simulation des procédés de traitements par lots appelés également systèmes batch [16]. Ces procédés, dans lesquels la matière est de nature continue et les traitements que subissent les lots de matière sont de type continu, présentent des aspects continus et discrets étroitement liés. Ils sont couramment rencontrés dans les industries de transformation de la matière.

Dans le domaine de la robotique, on considère un SDH constitué d'un contrôleur discret et d'un système de mouvements sous contraintes représentant la partie continue, ce système peut être un bras articulé ou d'autres types de robot. Ces deux parties communiquent par l'intermédiaire d'une interface et on utilise la structure modulaire pour représenter ce type de SDH [17].

On utilise le formalisme des automates hybrides pour des études sur la régulation du trafic aérien ainsi que sur la commande de véhicules sur des autoroutes automatisées. Dans [Tom98], les auteurs présentent une méthode pour résoudre les problèmes de conflits entre avions sur les lignes aériennes. Cette méthode passe par la modélisation des avions et de leurs différentes manoeuvres comme des systèmes de contrôle hybride. De même, la sécurité sur des autoroutes automatisées a fait l'objet de plusieurs études. Le système composé des différents véhicules et de l'autoroute est modélisé à l'aide de modules appelés plateaux regroupant plusieurs véhicules allant à la même vitesse et dans la même direction. Des ordres discrets donnés par le système de contrôle permettent aux véhicules de changer de plateau et d'avoir ainsi un mode de fonctionnement continu différent.

Dans le domaine de l'électronique de puissance, on s'intéresse à des systèmes dans lesquels l'ouverture ou la fermeture contrôlée ou autonome d'interrupteurs à semi-conducteurs provoque des modifications dans la structure du système [18]. On considère ces systèmes comme des SDH avec plusieurs modes de fonctionnement continu contrôlés par un SED qui commande les interrupteurs. On peut représenter ainsi ces systèmes à l'aide de la structure modulaire. On utilise également les réseaux de Petri pour modéliser la structure de commande. Un des problèmes de commande, qui peut être associé à ces systèmes, est le calcul des meilleurs instants de fermeture ou d'ouverture des interrupteurs pour permettre au système d'avoir un comportement désiré. On peut également chercher à déterminer quel est l'enchaînement optimum des différents modes de fonctionnement du système.

Après une introduction générale, notre travail a été réparti en trois chapitres fondamentaux, le premier porte sur des généralités et définitions permettant de se familiariser avec les systèmes hybrides, puis le second regroupe les trois points sur lesquels s'est portée notre étude, la stabilité, les observateurs et la commande, étude qui s'est le plus souvent accès sur une classe spécifique des systèmes hybrides, les systèmes à commutation de modèle. Afin d'illustrer les différentes approches exposées dans le chapitre deux, deux applications représentant toute deux des systèmes à commutation de modèle ont été données dans le chapitre trois, la première à caractère hydraulique est une régulation de niveau dans deux réservoirs, quand à la deuxième qui elle est à caractère électrique concerne la commande optimale avec contraintes du convertisseur abaisseur DC-DC. Dans le but de faciliter la compréhension de cette nouvelle notion d'hybridité, plusieurs exemples ont été donnés tout au long des chapitres. Pour enfin aboutir à une conclusion finale.

Introduction :

Nous donnons dans ce chapitre une définition et une classification des systèmes dynamiques hybrides. Nous présentons ensuite les différents moyens de représentation de ces systèmes. Puis, nous introduisons les outils de simulation des SDH. Enfin nous donnons des exemples qui illustrent les principales caractéristiques des SDH.

1.1 Définition des systèmes dynamiques hybrides :

Avant de donner une définition formelle des SDH, il convient de mieux situer ces systèmes par rapport aux systèmes purement discrets et purement continus. En effet, pour de nombreux systèmes appelés hybrides, l'aspect continu ou discret est tellement dominant que l'étude sous une approche hybride n'apporte rien de significatif. C'est pourquoi il nous semble nécessaire de déterminer les SDH pour lesquels cette étude sous une approche hybride est intéressante.

Une première définition des systèmes dynamiques hybrides est basée sur la notion de variable d'état :

Définition 1.1 :

Un système est dit hybride s'il possède deux types de variables d'état : des variables d'état continues et des variables d'état discrètes [19].

On peut dire, d'après la définition 1.1, que les systèmes échantillonnés et les systèmes discrets (voir annexe A) sont des systèmes dynamiques hybrides. Cependant, ces systèmes ne possèdent qu'une seule dynamique, continue ou discrète. Il est donc possible pour les étudier de les rendre purement discrets ou purement continus et d'appliquer les méthodes connues d'analyse des systèmes continus et discrets.

Les systèmes dynamiques hybrides que nous étudierons dans la suite de ce travail sont des systèmes composés de deux sous-systèmes : un système continu (SC) interagissant avec un système à événements discrets (SED). La communication entre les deux composantes s'effectue par l'intermédiaire d'interfaces.

1.1.1 Interactions entre les systèmes : phénomènes hybrides :

Nous allons maintenant étudier comment se produisent les interactions entre les deux sous systèmes (discret et continu) qui forment le SDH. Pour cela, nous allons d'abord présenter les actions discrètes qui peuvent intervenir lors de l'évolution d'un système continu décrit par un système différentiel de la forme :

$$\dot{x}(t) = \xi(t) \quad , t \geq 0$$

Où $x(t) \in \mathbb{R}^n$ est le vecteur d'état du système. On note τ l'instant où intervient une action discrète dont nous verrons ultérieurement comment elle peut être déclenchée. L'ensemble, action et déclenchement, est appelé **phénomène hybride** [20].

Actions des phénomènes hybrides :

- Les phénomènes hybrides agissent sur la dynamique du système continu, modifiant ainsi cette dynamique. Le système hybride se situe alors pour $t \geq \tau$ dans un autre mode de fonctionnement. On appelle ce phénomène **commutation/changement de modèle/dynamique** et τ **instant de commutation**. Le choix du nouveau modèle est déterminé par le SED. Un exemple simple de modèle formel avec deux modes de fonctionnement est le suivant :

$$\begin{aligned} \dot{x}(t) &= \xi_1(t) & , & & t < \tau \\ \dot{x}(t) &= \xi_2(t) & , & & t \geq \tau \end{aligned}$$

- Les phénomènes hybrides agissent également sur le vecteur d'état du système, le faisant évoluer de manière différente pour $t = \tau$. Une discontinuité apparaît alors sur le vecteur d'état. On appelle ce phénomène **saut de l'état**. Un exemple de modèle formel est représenté par les équations suivantes :

$$\begin{aligned} \dot{x}(t) &= \xi(t) & , & & t \geq 0 & \text{ et } t \neq \tau \\ \dot{x}(\tau) &= g(., \tau) \end{aligned}$$

- Les deux actions peuvent être couplées ; en effet on peut considérer qu'à l'instant τ , on a **une commutation de modèle et un saut du vecteur d'état**. Un exemple de modèle formel est le suivant

$$\begin{aligned} \dot{x}(t) &= \xi_1(t) & , & & t \leq \tau \\ \dot{x}(\tau) &= g(., \tau) \\ \dot{x}(t) &= \xi_2(t) & , & & t > \tau \end{aligned}$$

La fonction g provoque une discontinuité de l'état à l'instant τ , puis intervient la deuxième dynamique continue ξ_2 .

Déclenchements des phénomènes hybrides :

- Les phénomènes hybrides peuvent être déclenchés lorsque le vecteur d'état continu atteint certaines valeurs. L'événement ainsi créé fait évoluer le SED, celui-ci passe dans un nouvel état et suivant cet état une des actions décrites ci-dessus est générée sur le système continu. On dit alors que ce **phénomène est autonome**.
- Les phénomènes hybrides peuvent également être déclenchés par une commande discrète extérieure agissant sur le SED. On dit alors que ce **phénomène est contrôlé**.

1.1.2 Définition formelle des SDH et classification des phénomènes hybrides :

Définition formelle des SDH :

Nous proposons la définition formelle des systèmes dynamiques hybrides suivante :

Définition 1.2 [21]:

Soit $x(t) \in X \subset \mathbb{R}^n$, $q(t) \in Q \subset \mathbb{N}^m$, $u(t) \in U \subset \mathbb{R}^c$, $v(t) \in \Omega_c \subset \mathbb{N}^d$ et $t \in \mathbb{R}^+$ où n, m, c, d sont donnés.

L'ensemble X représente l'ensemble des états continus et l'ensemble Q représente l'ensemble des états discrets. L'ensemble U représente l'ensemble des commandes continues et l'ensemble Ω_c représente l'ensemble des commandes discrètes. La variable t représente le temps.

Soit $S = X \times Q$ et soit P un sous-ensemble fermé de $\mathbb{R}^n \times Q$. On définit l'état hybride $s(t) \in S$ du système à l'instant t par la donnée du couple $[x(t), q(t)]$ et on appelle SDH un système dynamique décrit par les équations suivantes pour $t \in [t_0, t_f]$:

$$\dot{x}(t) = F(x(t), q(t), u(t)) \quad \forall s(t) \in S \cap C_p \tag{1.2}$$

$$[x(t^+), q(t^+)] = G(x(t), q(t), u(t)) \quad \begin{cases} \text{si } s(t) \in \partial P \\ \text{si } v(t) \in \Omega_c \text{ intervient} \end{cases} \tag{1.3}$$

$$x(t_0) = x_0 \tag{1.4}$$

$$q(t_0) = q_0 \tag{1.5}$$

La fonction F représente la dynamique continue du système hybride et G représente la dynamique hybride du système correspondant aux phénomènes hybrides décrits précédemment. L'ensemble P est celui des zones de déclenchement des phénomènes hybrides et ∂P est la frontière de P . La commande discrète $v(t)$ détermine les instants où la fonction G intervient (instants de commutation/saut). Elle détermine également le nouveau modèle ou l'amplitude du saut de l'état suivant l'action du phénomène hybride. Les fonctions F et G sont supposées suffisamment régulières pour que le système défini par les équations (1.2-1.5) admette une solution unique.

Classification des phénomènes hybrides [21] :

Dans cette partie, nous proposons une classification des phénomènes hybrides intervenant dans les SDH, sachant qu'un SDH peut posséder plusieurs types de phénomènes. Ces phénomènes étant décrits par la fonction G , on définit les différents phénomènes hybrides suivant les propriétés de cette fonction G et notamment suivant son ensemble de départ, noté D et son ensemble d'arrivée, noté A .

Le tableau 1.1 montre, suivant l'ensemble de départ de la fonction G , le type de déclenchement correspondant.

$D=X \times Q$	$D=\Omega_c$
Déclenchement Autonome	Déclenchement Contrôlé

Tableau 1.1 : Déclenchements des phénomènes hybrides.

Le tableau 1.2 montre suivant l'ensemble d'arrivée de la fonction G , le type d'action correspondant.

$A=Q$	$A=X$
Commutation de Modèle	Saut de L'état

Tableau 1.2 : Actions des phénomènes hybrides.

Selon leur action et la manière dont ils sont déclenchés, on distingue quatre types de phénomènes hybrides [Bra95] :

- Les **Commutations de modèle** Autonomes, notées G_{CA} .
- Les **Commutations de modèle** contrôlées, notées G_{CC} .
- Les **Sauts de l'états** Autonomes, notés G_{SA} .
- Les **Sauts de l'états** Contrôlés, notés G_{SC} .

Ces quatre phénomènes sont représentés graphiquement sur la figure 1.1.

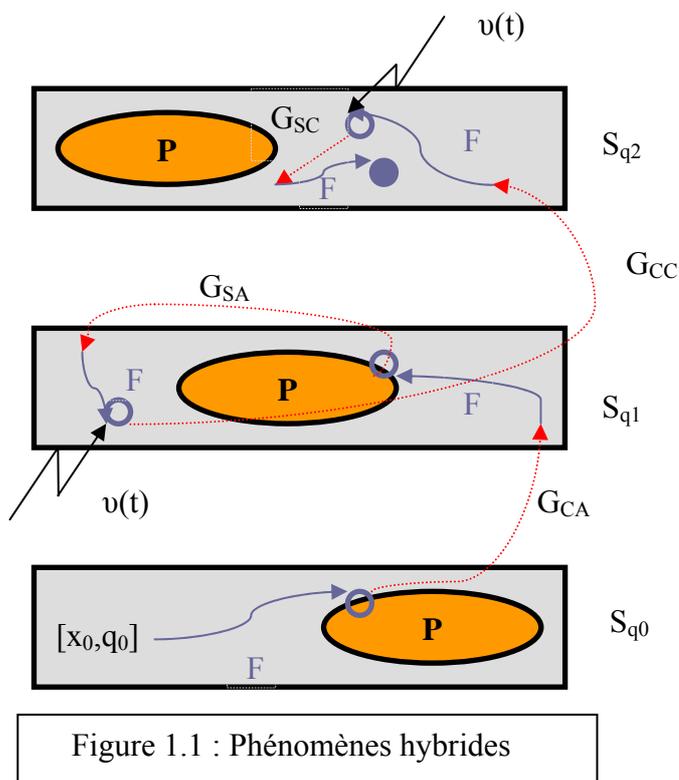


Figure 1.1 : Phénomènes hybrides

Où $S_{q_0} = (X, q_0) \subset S$, $S_{q_1} = (X, q_1) \subset S$ et $S_{q_2} = (X, q_2) \subset S$ avec $Q = \{q_0, q_1, q_2\}$.

Ainsi, en partant de l'état initial $s(t_0) = [x_0; q_0] \in S_{q_0}$, le système évolue suivant la dynamique F jusqu'à ce que s atteigne l'ensemble P . A cet instant noté τ_1 , la dynamique hybride intervient et modifie la valeur de q qui devient égale alors à q_1 . Ce phénomène est donc une commutation autonome (G_{CA}).

Ensuite, le système poursuit son évolution avec une nouvelle dynamique définie par F (qui a été modifiée). À l'instant τ_2 , s atteint à nouveau l'ensemble P , mais cette fois la dynamique hybride modifie x . Ce phénomène est donc un saut autonome (G_{SA}).

Enfin, ces deux phénomènes se reproduisent mais sur l'intervention d'une commande discrète $v(t)$ qui détermine dans un cas le nouveau modèle (G_{CC}) et dans l'autre cas la nouvelle valeur du vecteur d'état (G_{SC}).

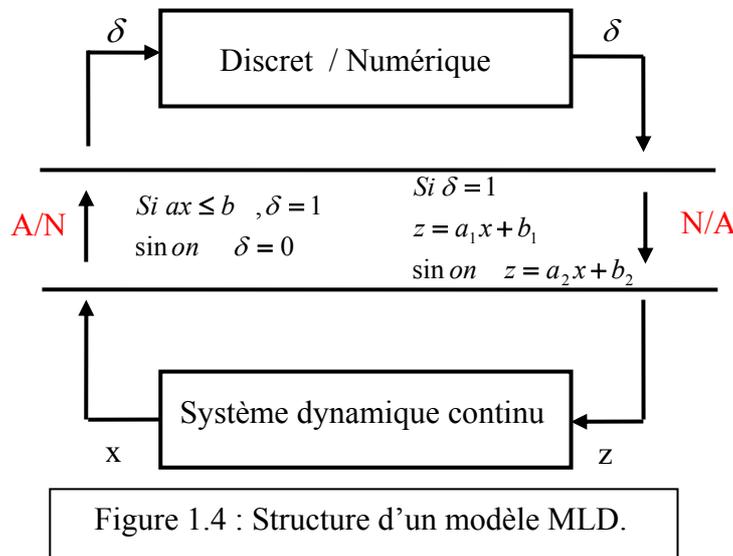
1.2 Représentations des systèmes dynamiques hybrides :

Les moyens de représentation des SED couramment utilisés sont les réseaux de Pétri (RdP) et les automates. Des travaux ont déjà été réalisés pour étendre les réseaux de Pétri et les automates aux SDH, qui sont utilisés pour modéliser les SDH sous un seul formalisme, principalement dans le but de simuler et d'analyser ces systèmes complexes.

Il existe aussi une modélisation dite modulaire dans laquelle les deux dynamiques continue et discrète sont séparées, la communication entre ces deux parties s'effectue par l'intermédiaire d'une interface. Enfin nous présentons le formalisme MLD et PWA.

1.2.1 Modèle mixte dynamique et logique « Mixed Logical Dynamical » (MLD) :

La forme MLD représente le système hybride par des équations linéaires, avec des inégalités linéaires portant sur les variables continues et binaires. L'ensemble des inégalités de la forme MLD rassemble donc d'une part les contraintes du système, d'autre part les inégalités issues des propositions logiques et des variables auxiliaires.



Des variables auxiliaires permettent de modéliser les relations existantes entre les parties continue et discrète. Ainsi, le passage de la partie continue à la partie discrète nécessite l'ajout de variables logiques[7].

Les systèmes sous formalisme MLD résultant de l'association continu/discret sont dès lors décrits par les relations linéaires suivantes :

$$x(k+1) = Ax(k) + B_1u(k) + B_2\delta(k) + B_3z(k) \quad (1.9)$$

$$y(k) = Cx(k) + D_1u(k) + D_2\delta(k) + D_3z(k) \quad (1.10)$$

$$E_2\delta(k) + E_3z(k) \leq E_1u(k) + E_4x(k) + E_5 \quad (1.11)$$

Cette relation fait intervenir les grandeurs suivantes, mixtes continues/binaires, ou binaires, ou continues :

$$\text{L'état : } x = \begin{bmatrix} x_c \\ x_l \end{bmatrix}, x_c \in R^{n_c}, x_l \in \{0,1\}^{n_l} \text{ et } n = n_c + n_l$$

$$\text{La sortie : } y = \begin{bmatrix} y_c \\ y_l \end{bmatrix}, y_c \in R^{p_c}, y_l \in \{0,1\}^{p_l} \text{ et } p = p_c + p_l$$

$$\text{L'entrée : } u = \begin{bmatrix} u_c \\ u_l \end{bmatrix}, u_c \in R^{m_c}, u_l \in \{0,1\}^{m_l} \text{ et } m = m_c + m_l$$

$$\text{Les variables binaires auxiliaires : } \delta \in \{0,1\}^r$$

$$\text{Les variables continues auxiliaires : } z \in R^{r_c}$$

Un problème est dit « bien posé » si la solution en δ et z est unique pour une paire donnée (u, x) et par conséquent $x(k+1)$ est défini de façon unique, ce qui permet de trouver la trajectoire unique des états x .

1.2.2 Modèle affine par morceaux (Piecewise Affine System PWA) :

L'évolution d'un système peut être gouvernée par différents modèles affines selon les valeurs courantes des états et des entrées. Dès lors, l'espace global est divisé en plusieurs régions, chaque région étant définie par des inégalités sur les états et les entrées. La dynamique de chaque région est caractérisée par une dynamique affine particulière.

Un modèle affine par morceaux est défini de la façon suivante :

$$S^i : \left\{ \begin{array}{l} x(k+1) = A^i x(k) + B^i u(k) + f^i \\ y(k) = C^i x(k) + g^i \end{array} \right\}, \text{ pour } \begin{bmatrix} x(k) \\ u(k) \end{bmatrix} \in \chi_i \quad (1.12)$$

Où $\{\chi_i\}_{i=1}^s$ représente la partition polyédrale de l'espace d'état et de commande (s est le nombre d'éléments de cette partition).

Chaque χ_i est donné par :

$$\chi_i = \left\{ \begin{bmatrix} x(k) \\ u(k) \end{bmatrix} \middle| Q^i \begin{bmatrix} x(k) \\ u(k) \end{bmatrix} \leq q^i \right\} \quad (1.13)$$

Où $x(k), u(k), y(k)$ sont respectivement les vecteurs d'état, de commande et de sortie à l'instant k . Chaque sous-système S^i défini par le 7-uplet $(A^i, B^i, C^i, f^i, g^i, Q^i, q^i), i \in \{1, 2, \dots, s\}$ est un composant du modèle PWA. $A^i \in \mathbb{R}^{n \times n}, B^i \in \mathbb{R}^{n \times m}, C^i \in \mathbb{R}^{p \times n}, Q^i \in \mathbb{R}^{p_i(n+m)}, f^i, g^i, q^i$ sont des vecteurs constants de dimensions appropriées, où n, m, p sont respectivement les dimensions de l'état, de l'entrée, de la sortie, et p_i est le nombre d'hyperplans définissant le $i^{\text{ème}}$ polyèdre. Dans ce formalisme, une entrée de commande logique est prise en compte en développant un modèle affine pour chaque valeur 1 et 0 de cette grandeur, et en définissant des contraintes linéaires de type inégalité liant le modèle avec la valeur correspondante de cette entrée.

Il a été montré dans [22], que les modèles MLD et les modèles PWA sont équivalents, ce qui permet le passage d'un modèle à l'autre.

La figure ci-dessous illustre une répartition polyédrale dans le plan :

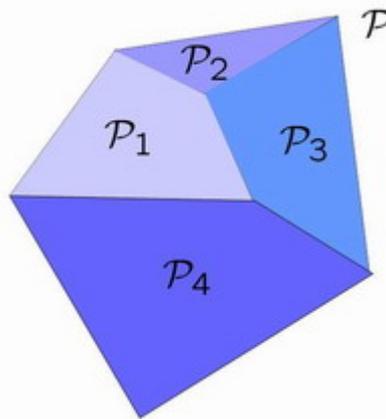


Figure 1.5 : Exemple d'une répartition polyédrale dans le plan

1.3 Outils de simulation :

Les simulations des modèles hybrides posent des problèmes spécifiques liés à l'interaction entre les états discrets et les états continus. Seuls quelques systèmes hybrides admettent des solutions analytiques. Les systèmes hybrides linéaires ou affines peuvent être résolus analytiquement, mais les efforts pour trouver les solutions augmentent avec l'ordre de l'espace continu, le nombre de variables discrètes et la complexité des frontières de commutations. Une multitude d'algorithmes utilisant ces méthodes numériques sont disponibles. Ils permettent de représenter n'importe quel système hybride et de le simuler. Les plus connus sont Simulink combiné avec Stateflow (Annexe ?); Scicos (l'équivalent en logiciel libre) [23][24] ; Omola [25] ; Dymola [26] ; Modelica [27] et HyBrSym [28], et une nouvelle boîte à outil de multiparamétrique de Matlab (Annexe E).

Dans notre cas nous avons utilisé Simulink combiné avec Stateflow et la boîte à outil multi paramétrique de Matlab, le premier est un outil de conception de systèmes événementiels. Basé sur la théorie des machines à états finis, il permet de concevoir graphiquement des systèmes à événements discrets et de simuler leur fonctionnement. Entièrement intégré à Simulink, il complète parfaitement cet environnement de simulation, quand à la deuxième récemment mise au point dans un laboratoire suisse, elle permet de modéliser d'analyser de commander et de représenter les systèmes hybrides avec toutes leurs dynamiques et caractéristiques comme aucun logiciel auparavant.

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

- simuler le fonctionnement de la partie discrète ;
- en déduire le ou les modèles actifs (modes) pour la partie continue ;
- résoudre les équations différentielles ou aux différences constituant cette partie continue.

1.4. Exemples de systèmes hybrides:

1.4.1 La balle rebondissante :

La balle rebondissante est un système basé sur le phénomène de saut d'état qui décrit l'évolution d'une variable continue $x(t)$ régie par une équation différentielle sous contraintes. Lorsque $x(t)$ vérifie certaines conditions, il est alors possible de lui donner une impulsion, c'est à dire de lui affecter une nouvelle valeur.

On considère une balle de masse m soumise à l'action de la gravité. On la laisse tomber d'une altitude z_0 avec une vitesse initiale nulle. L'altitude $z(t)$ de la balle suit donc l'équation différentielle issue de la mécanique classique :

$$m \ddot{z}(t) = -mg$$

Quand $z(t) = 0$, la balle touche le sol et rebondit en perdant une fraction de son énergie :

$$\ddot{z}(t^+) = -c \ddot{z}(t^-) \quad , \text{ avec } c \leq 1.$$

Où c est le coefficient de restitution.

En posant $x_1(t) = z(t)$, $x_2(t) = \dot{z}(t)$ et en utilisant le formalisme des automates hybrides, le modèle hybride de la balle rebondissante est donné par :

1. $Q = \{q\}$
2. $E = \{e = (q, q)\}$
3. $D_q : R^+ \rightarrow R$
4. $f_q(x_1, x_2) = (x_2, -g)$
5. $G_e = \{x_1 = 0\}$
6. $\mathfrak{R}_e(x_1, x_2) = \{(x_1, -cx_2)\}$

L'automate hybride de la balle rebondissante est représenté ci-dessous :

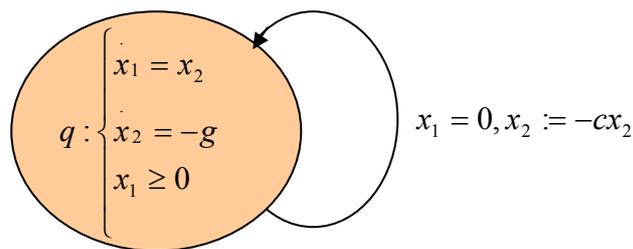


Figure1.6 : Automate hybride de la balle rebondissante

La balle rebondissante est un exemple des systèmes Zénon c'est-à-dire le système exécute en un temps fini un nombre infini de commutations.

Pour mieux voir ce type de système hybride, système hybride avec sauts d'états (Resets), des simulations sont effectuées sur le modèle de la balle rebondissante pour des conditions initiales $x_1=5$ m, $x_2=0$ m/sec et un coefficient de restitution $c=0.3$, les deux variables d'états sont visualisées ainsi que l'événement x_1 négative, qui détermine l'instant du saut.

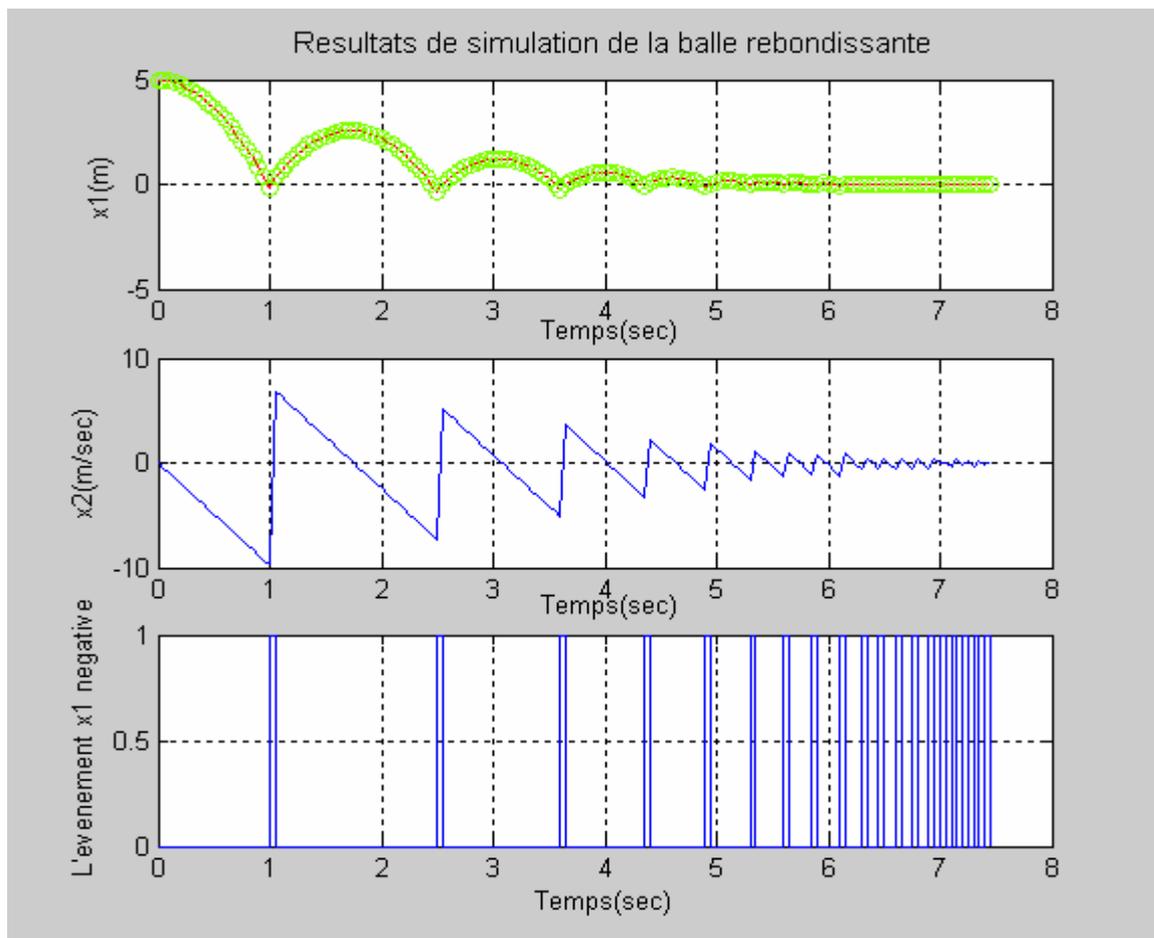


Figure1.7 : Simulation de comportement de la balle rebondissante

1.4.2 Thermostat :

Exemple simple et classique de SDH. En effet, de nombreuses approches proposées pour la description et l'analyse des SDH sont souvent illustrées à l'aide d'un système dont la fonction est de maintenir constante la température d'un local (la variable x). Le régulateur de température utilisé est le thermostat à action tout ou rien modélisé par le système d'état suivant :

$$\dot{x} = \begin{cases} -x + 4 & x \leq \theta_M \text{ on} \\ -x & x \geq \theta_m \text{ off} \end{cases}$$

Tels que θ_M et θ_m sont les seuils de température, fixés comme limite inférieure et supérieure de la température de la pièce.

On obtient ainsi l'automate hybride H représenté sur la figure :

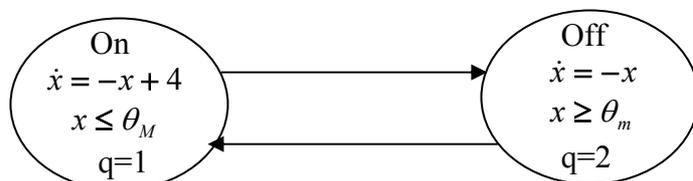


Figure 1.8 : L'automate hybride du thermostat

La variable discrète q prend ses valeurs en fonction des deux seuils de température. Les variations de la température de la pièce et de la variable q sont représentées sur la figure 1.12. Pour les simulations on a choisi $\theta_M = 20^\circ\text{C}$ et $\theta_m = 18^\circ\text{C}$ et une condition initiale de 19°C .

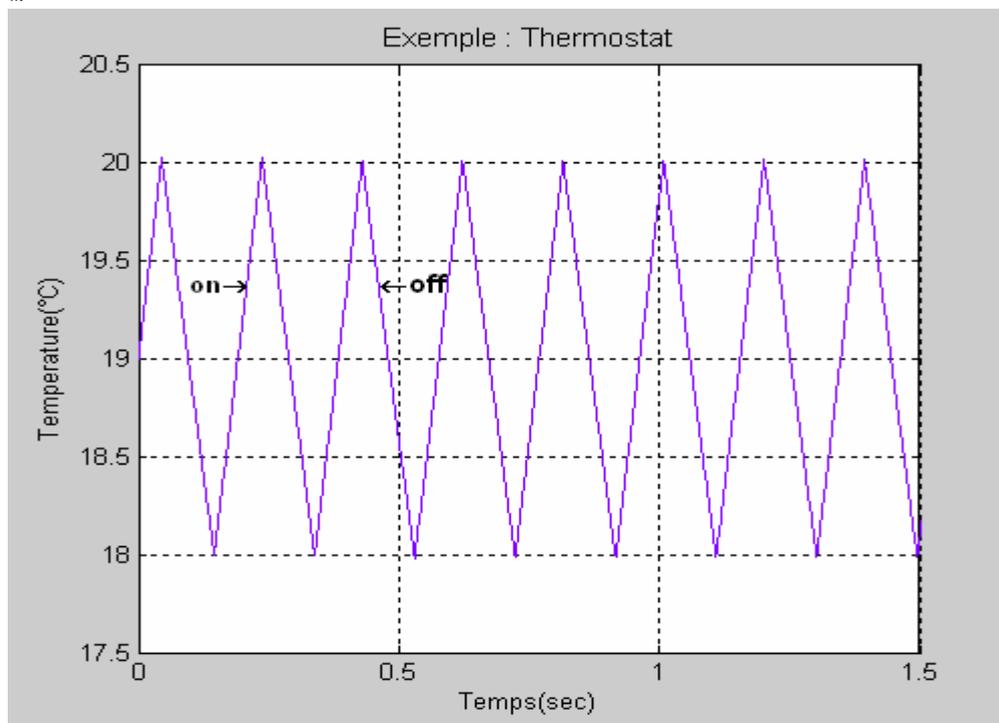


Figure 1.9.a : L'évolution de la température d'un local

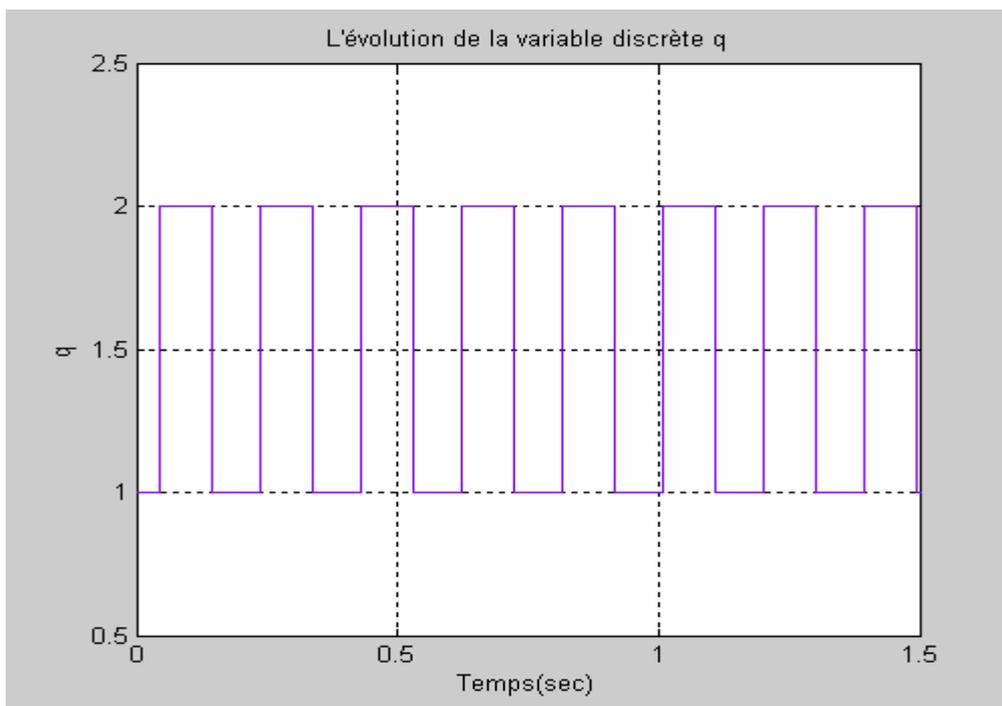
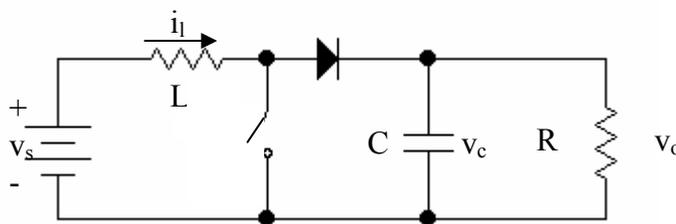


Figure 1.9.b : L'évolution de la variable discrète q

Figure 1.9 : Comportement de la boucle de régulation de température

1.4.3 Le Convertisseur Boost :

Une classe particulièrement importante en terme d'applications est formée par les convertisseurs de puissance, couramment embarqués sur les ordinateurs, les moteurs électriques, les téléphones portables et plus généralement sur tous les dispositifs comportant une alimentation. Les convertisseurs DC-DC présentent par leur nature un comportement hybride. L'aspect hybride est décrit par la présence des variables discrètes (états des interrupteurs) et des variables continues (courants et tensions). On se contentera ici de présenter le comportement en boucle ouverte d'un élévateur de tension (Boost), le montage ci-dessous permet de fournir une tension de sortie v_0 supérieure à la tension d'entrée v_s .



Quand l'interrupteur est fermé, la diode est polarisée en inverse la charge est donc isolée de la source, on obtient les équations :

$$\frac{di_l(t)}{dt} = \frac{1}{L} v_s(t)$$

$$\frac{dv_c(t)}{dt} = -\frac{1}{RC} v_c(t)$$

Quand l'interrupteur est ouvert, la diode conduit la charge donc reçoit de l'énergie de la source, on obtient ainsi les équations :

$$\frac{di_l(t)}{dt} = -\frac{1}{L} v_c(t) + \frac{1}{L} v_s(t)$$

$$\frac{dv_c(t)}{dt} = \frac{1}{C} i_l - \frac{1}{RC} v_c(t)$$

Ces deux modes de fonctionnement représentent respectivement les états discrets 1=Son et 2= Soff de l'interrupteur. Un troisième mode 3=Snc peut être ajoutée pour obtenir un modèle hybride plus précis, c'est le mode de non conduction c'est-à-dire dans le cas où l'interrupteur est ouvert , avec i_l positif , la tension aux bornes de la capacité augmente et le courant i_l diminue ,et si le temps t_{off} est suffisant ,au moment où i_l s'annule la capacité tente à se décharger à travers la diode ce qui est impossible donc le courant est nulle . Dans ce cas, avec $i_l=0$, le montage se réduit à une seule maille qui comporte R et C, ce circuit est décrit par les équations :

$$\frac{di_l(t)}{dt} = 0$$

$$\frac{dv_c(t)}{dt} = -\frac{1}{RC} v_c(t)$$

Le modèle comporte ainsi trois états discrets, assortis de trois équations différentielles.

Le modèle sous forme d'état est donné par :

$$\dot{x}(t) = Ax(t) + Bu(t)$$

Les matrices A et B sont données dans le tableau suivant :

	A	B
Son	A2	B1
Soff	A1	B1
Snc	A2	B2

Tels que :

$$A1 = \begin{bmatrix} 0 & \frac{-1}{L} \\ \frac{1}{C} & \frac{-1}{RC} \end{bmatrix}, B1 = \begin{bmatrix} \frac{1}{L} \\ 0 \end{bmatrix} ;$$

$$A2 = \begin{bmatrix} 0 & 0 \\ 0 & \frac{-1}{RC} \end{bmatrix}, B2 = \begin{bmatrix} 0 \\ 0 \end{bmatrix} ;$$

La simulation a été faite sous Stateflow /Simulink , le diagramme (Flowchart) de boost est donné par :

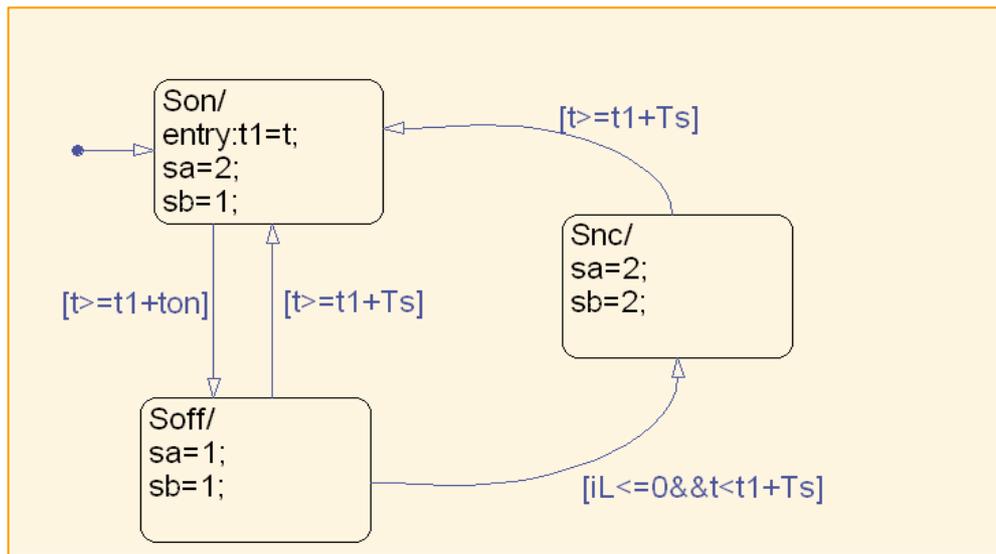


Figure 1.10 : Flowchart de Boost

Tel que ton est le temps où l'interrupteur est fermé et Ts est la période de découpage. Le schéma global de simulation du convertisseur est donné ci-dessous :

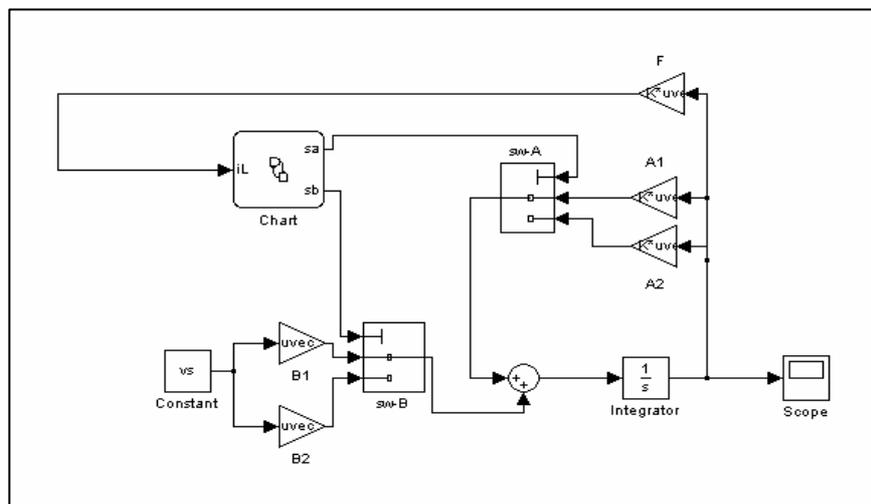


Figure1.11 : Schéma complet de simulation du convertisseur Boost

Les simulations sont exécutées pour les valeurs : $Ts=5 \text{ e-}4 \text{ sec}$, $ton = 1.2\text{e-}4 \text{ sec}$, $vs=20 \text{ volt}$.
 $R=4 \text{ ohm}$, $L=250\text{e-}6 \text{ H}$, $C=100\text{e-}6 \text{ F}$.

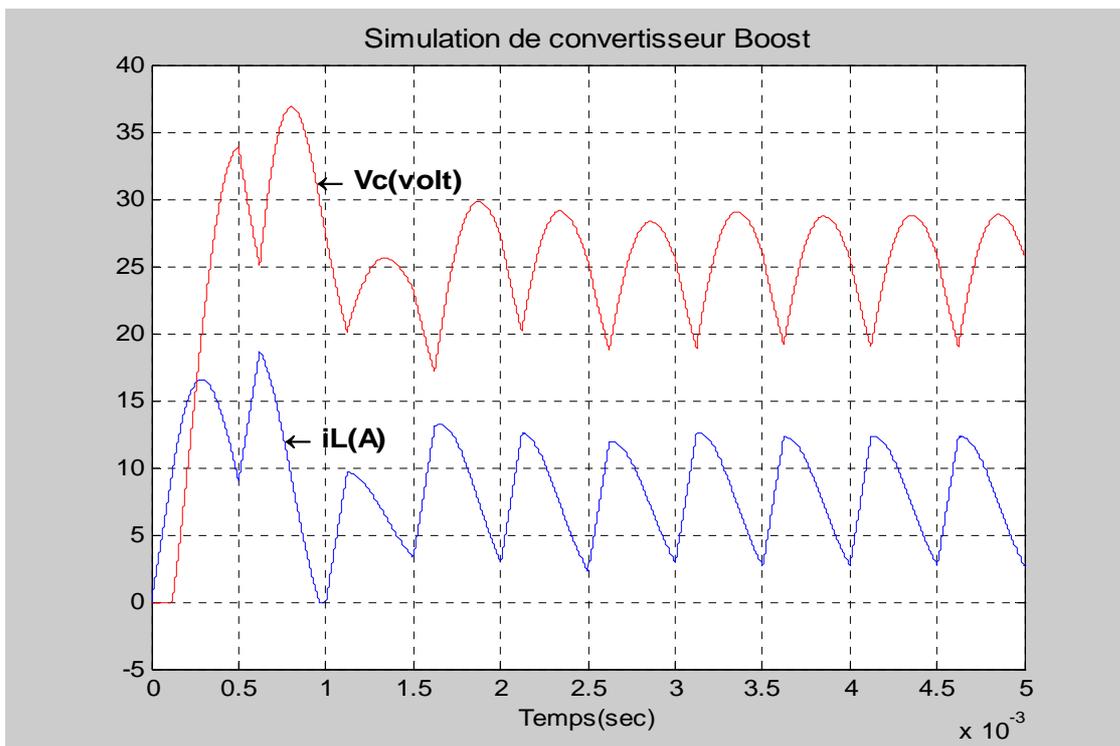


Figure1.12 : Résultats de simulation de convertisseur Boost en mode de fonctionnement continu

En fin si on change la valeur de la résistance $R=10$ ohm le mode discontinue apparaît (c'est-à-dire le mode de non conduction).

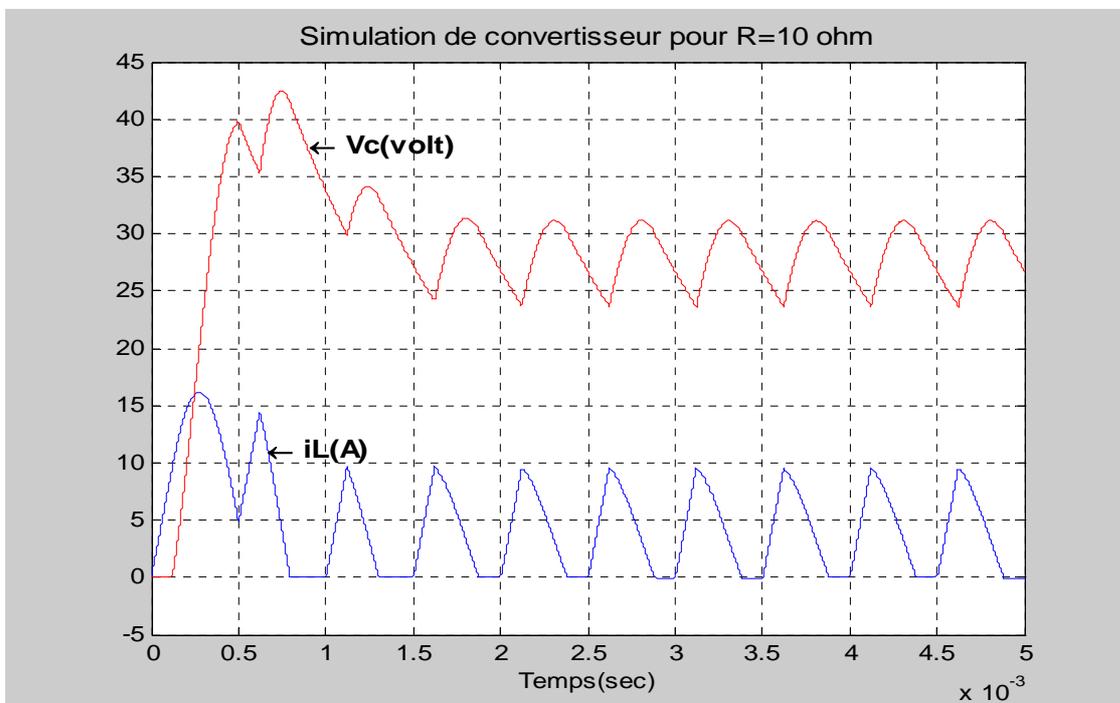


Figure1.13 : Résultats de simulation de convertisseur Boost en mode de fonctionnement discontinu

Conclusion :

Ce chapitre nous a permis de mieux situer les SDH par rapport aux systèmes continus et aux systèmes à événements discrets grâce notamment à une analyse des phénomènes hybrides. Une définition formelle a été donnée et différents moyens de représentation des SDH ont été présentés. Nous avons proposé également une classification des SDH. Ensuite les outils de simulation des SDH ont été exposés. Enfin des exemples pris dans des domaines variés ont illustrer les caractéristiques des SDH.

Introduction :

Ayant suscité l'intérêt de plusieurs chercheurs par leurs caractéristiques intrinsèques cette dernière décennie, les travaux sur les systèmes hybrides ont été nombreux, on a choisi de présenter dans ce chapitre les principaux d'entre eux dans trois domaines fondamentaux la stabilité, l'observation et la commande.

1. La stabilité :

La théorie classique de la stabilité de Lyapunov a été la base de la stabilité des systèmes au siècle dernier, où la déduction de la stabilité dépend de l'existence et /ou la construction d'une fonction continue dérivable de Lyapunov appropriée. Qui peut ne pas exister ou être difficile à construire quand elle existe. Pour les systèmes hybrides la démonstration de l'existence ou la construction d'une fonction de Lyapunov devient encore plus difficile.

Les deux exemples suivants donnent une idée des défis ou difficultés posés par les caractéristiques inattendus et intrinsèques des S.D.H.

Exemple 1 :

On considère les dynamiques d'état autonomes suivantes :

$\dot{x}(t) = A_p x(t)$ Ou $x = [x_1 \ x_2]^T \in \mathbb{R}^2$ et $p \in \{1,2\}$ et :

$$A_1 = \begin{bmatrix} -1 & -100 \\ 10 & -1 \end{bmatrix}; A_2 = \begin{bmatrix} -1 & 10 \\ -100 & -1 \end{bmatrix}$$

, où les deux matrices A_1 et A_2 dont les valeurs propres sont : $\lambda_{1,2} = -1 \pm j\sqrt{1000}$, sont stables. On définit la séquence de commutation P comme suit :

$$P(t^+) = \begin{cases} 1 & \text{if } P(t) = 2 \text{ et } x_2(t) = (-1/k) * x_1(t) \\ 2 & \text{if } P(t) = 1 \text{ et } x_2(t) = k * x_1(t) \end{cases} \quad (2.1)$$

Pour des valeurs initiales quelconques, cette fonction détermine les conditions de commutation de la dynamique du système entre A_1 et A_2 en se basant sur une mémoire de l'état précédent. Pour $k=-0.2$ et $\forall x(0) \neq 0$; la trajectoire d'état diverge vers l'infini (figure 1.b). Ce qui démontre que la commutation entre deux systèmes A .stables, dont les diagrammes de phase sont donnés dans la figure 1.a ne donne pas forcément un système stable.

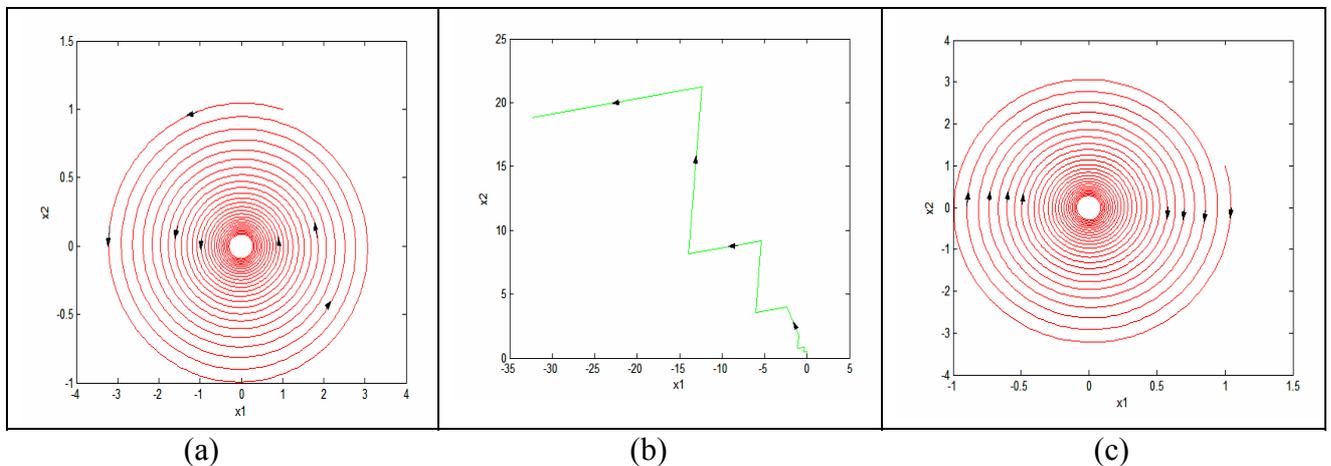


Figure 2.1.a et 2.1.c: Diagramme de phase stable de A_1 et A_2
Figure 2.1.b : Diagramme de phase instable

Cet exemple nous amène à poser deux questions fondamentales :

1. Quelle classe de systèmes stables admet une trajectoire d'état stable pour toutes les séquences de commutation ?
2. Quelles sont les séquences de commutation qui induisent toujours des trajectoires d'états stables ?

Un autre phénomène encore plus complexe, dual à celui présenté dans l'exemple2 est la commutation entre deux systèmes instables afin de produire une trajectoire finale stable.

Exemple 2: [32], [33]

Considérons les deux dynamiques autonomes :

$$\dot{x}(t) = A_{p(t)}x(t) \text{ ou } p(t) \in \{1,2\} \text{ et :}$$

$$A_1 = \begin{bmatrix} 0 & 10 \\ 0 & 0 \end{bmatrix}; A_2 = \begin{bmatrix} 1.5 & 2 \\ -2 & -0.5 \end{bmatrix} \tag{2.2.a}$$

Et la séquence de commutation :

$$p(t) = \begin{cases} 1; & \text{si } p(t^-) = 2 \text{ et } x_2(t) = -0.25x_1(t) \\ 2; & \text{si } p(t^-) = 1 \text{ et } x_2(t) = 0.5x_1(t) \end{cases} \tag{2.2.b}$$

Comme on peut le voir sur la figure 2.a , les deux systèmes sont instables tel que les valeurs propres de A_1 sont nulles et celles de A_2 $0.5 \pm j\sqrt{3}$.

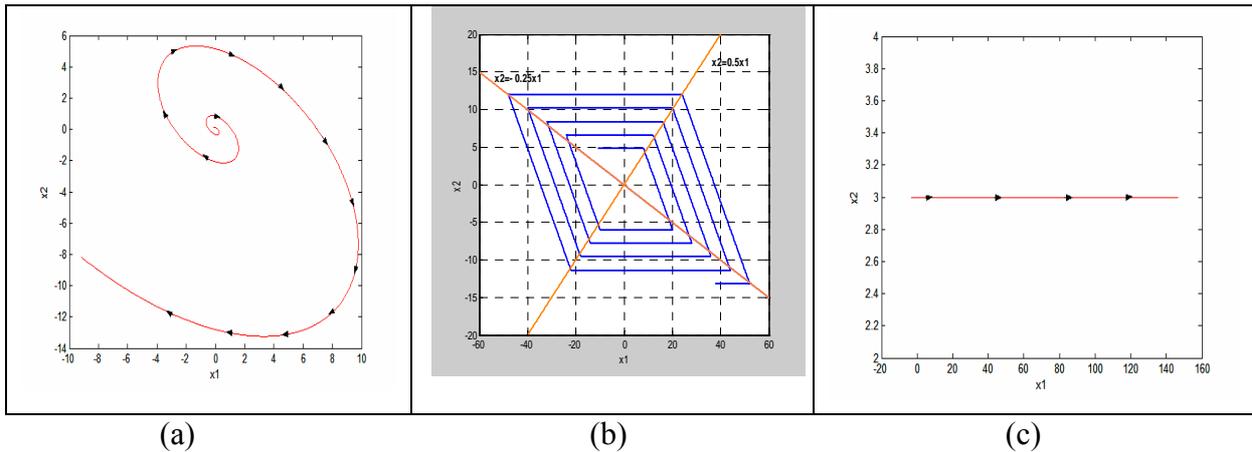


Figure 2.2.a et 2.2.c : Diagramme de phase instable de A1 et A2
Figure 2.2.b : Diagramme de phase stable

Les deux trajectoires de phase instables peuvent être rassemblées pour produire une trajectoire de phase stable représentée à la figure 1.b. Une étude détaillée est consacrée à ce type de système plus loin dans ce chapitre.

1.1. Résultats généraux sur la stabilité des S.D.H. :

La nature intrinsèque discontinue des S.D.H. a amené les chercheurs à penser à assembler les fonctions de Lyapunov Like des sous systèmes constituant le S.D.H afin d'aboutir à une fonction de Lyapunov non orthodoxe continue, quadratique par morceaux satisfaisant les conditions de stabilité [34] ;[35] ; [36], [37], [37], [38].

Dans la plupart des travaux sur la stabilité des S.D.H n'a été considéré que le cas ou le système est à commutation de modèle sous la forme :

$$\dot{x} = f_i(x) \tag{2.3}$$

, en ajoutant les deux conditions suivantes [39] :

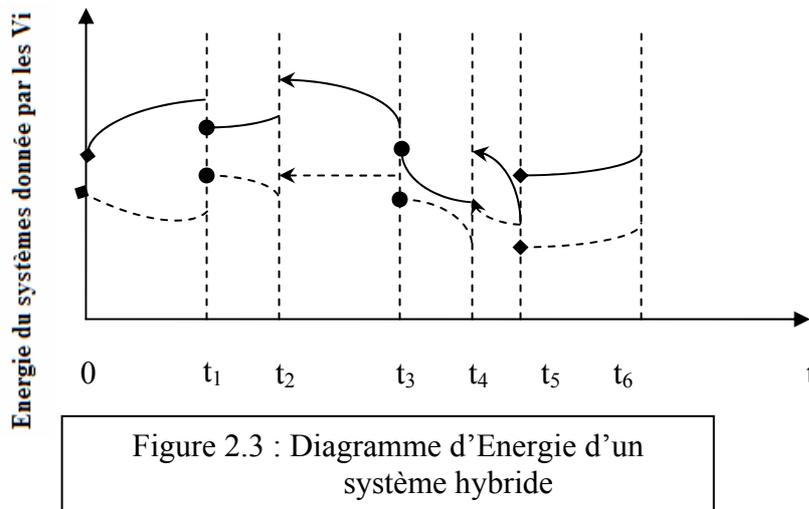
1. Chaque f_i est continue selon Liptchiz.
2. la séquence de commutation est choisie tel qu'il y ait un nombre fini de commutations en un temps fini.

Théorème 2 : (MLF) [40], [41] :

Soit le système non linéaire à M commutations (2.3) .On suppose qu'à chaque champ de vecteur f_i est associée une fonction de Lyapunov Like $V_i(x)$ dans le domaine Ω_i ; dont le point d'équilibre est $\bar{x} = 0$ et on suppose que $\bigcup_i \Omega_i = \mathbb{R}^n$.Soit P(t) une séquence de commutation donnée tel que : P(t) peut prendre la valeur i seulement si $x(t) \in \Omega_i$ et :

$$V_i(x(t_{i,k})) \leq V_i(x(t_{i,k-1})) \tag{2.4}$$

Alors le système (2.3) est *Lyapunov stable*.



La figure 2.3 décrit explicitement le sens de l'expression (2.4), et un résultat plus général exposé par Ye et al.([42],[43],[44]).Les résultats généraux présentés ci dessus fournissent des conditions suffisantes pour la stabilité des systèmes hybrides ;qui conduisent souvent à des règles de conception efficaces .

1.2. Stabilité et stabilisation des systèmes hybride linéaires à commutation de modèle :

Cette partie étudie quelques résultats importants appliqués au M-switched system .c à d qui vérifient :

$$\dot{x}(t) = A_{p(t)}x(t) \tag{2.5.a}$$

$$p(t^+) = \varphi(x(t), p(t)) \tag{2.5.b}$$

tel que : $p(t) \in \{1,2,\dots,\dots, M\}$. Pour cette classe spécifique de systèmes hybrides il existe des résultats plus efficaces que ceux décrit précédemment permis par Des choix particuliers de la fonction de commutation (2.5.b).

a) **Commutation entre des matrices A_i stables :**

Supposons que les $A_i; i = 1, \dots, M$. soient stables ç à d que toutes leurs valeurs propres sont dans la parties gauche du plan complexe.

Si tous les A_i ont la même fonction de Lyapunov $V(x) = x^T P x$ tel que :

$$\dot{V}(x) = -x^T Q x, \quad Q > 0$$

Alors le système est *exponentiellement stable* pour n'importe qu'elle séquence de commutation. Les conditions ci-dessus sur $V(x)$ sont équivalentes à $A_i^T P + P A_i \leq -Q$ pour tout i . Se référer à [47],[48] pour plus de détails.

b) **Commutation entre des matrices A_i quelconques :**

Si au moins une matrice $A_i \in A = \{A_1, \dots, A_M\}$ possède des valeurs propres dans la partie droite du plan complexe, alors il existe une séquence de commutation $p(t)$ capable de déstabiliser le système (2.3). C'est pourquoi l'existence et la construction d'une séquence de commutation stabilisante deviennent extrêmement importantes .

Quand, $M=2$, c à d, $A = \{A_1, A_2\}$, alors les corollaires de [32],[49] et [50] donnent :

Théorème 4:

Il existe une séquence de commutation $p(t)$ assurant la stabilité quadratique du système s'il existe $\alpha \in [0,1]$ tel que :

$$A_{eq} = \alpha A_1 + (1 - \alpha) A_2, \quad (2.6)$$

soit une matrice stable, ç à d que ses valeurs propres soient strictement dans la partie gauche du plan complexe.

Les conditions pour trouver A_{eq} sont données dans [46], [51],[52] satisfaisant :

Théorème 6 :

Soit $A = \{A_1, \dots, A_M\}$. Une condition suffisante pour l'existence d'une séquence de commutation $p(t)$ produisant un état $x(t)$ stable vérifiant l'équation du système (2.3) est l'existence d'une combinaison convexe stable des matrices A_i c à d qu'il existe $\alpha_i > 0, \sum_i \alpha_i = 1$:

$$A_{eq} = \sum_{i=1}^{i=M} \alpha_i A_i \quad (2.7)$$

Soit une matrice stable.

La question des conditions de structuration générales des matrices A_i garantissant la stabilité reste toujours posé. Des travaux effectués dans ce sens peuvent être trouvés dans [38] et [43]-[45]. Le résultats le plus important est l'existence d'une solution aux équations de Lyapunov tel qu'il est démontré dans [53].

2.1.3. Les conditions de stabilité sous forme d'inégalité matricielle linéaire :

Le principal défi dans les applications pratiques des systèmes hybrides est de trouver une fonction de Lyapunov vérifiant les conditions de stabilité. Malheureusement, il n'existe pas de méthodes systématiques .Toute fois pour les systèmes linéaires à commutation de modèle tel que ceux étudié dans la partie précédente, il existe un « LMI problem formulation» pour la construction d'un ensemble de fonction de Lyapunov like quadratique [54],[56] ,[57].

L'existence d'une solution pour le « LMI problem » est une condition suffisante à la stabilité d'un système hybride. Le grand avantage de cette approche est que « LMI problem » possède des solutions numériques efficaces et sûres obtenues par des techniques standard [56],[57].

Dans le cas ou on désire garantir la stabilité de toutes les séquences de commutations pour des matrices A_i asymptotiquement stable, alors le « LMI problem » est de trouver la matrice P tel que : $P = P^T > \mathbf{0}$ et $A_i^T P + A_i P < \mathbf{0}$, ([55],[58])

En suivant les étapes décrites dans [ref] la stabilité exponentielle des systèmes hybrides avec des champs de vecteur linéaire peut être vérifié par le « LMI problem » suivant :

LMI problem :

Soit I la matrice identité .S'il existe une solution à **min** β tel que :

1. $\alpha > \mathbf{0}, \mu_q \geq \mathbf{0}, \nu_q \geq \mathbf{0}, \zeta_q \geq \mathbf{0}, q \in \{1, \dots, k\}$.
2. $\alpha I + \mu_q Q_q \leq P_q \leq \beta I - \nu$
3. $A_i^T P_q + P_q A_i + \zeta_q Q_q \leq -I, (q, i) \in D_1$
4. $P_r + \eta_{q,r} Q_{q,r} \leq P_q, (q, r) \in D_2$

, ou k représente le nombre de fonction de Lyapunov like locale utilisées Alors :

Le point d'équilibre zéro est *exponentiellement stable dans le sens de Lyapunov*.

L'inégalité gauche de la deuxième condition, exige la définitude et la positivité des fonctions de Lyapunov like. L'inégalité droite est introduite afin de trouver une borne supérieure des fonctions de Lyapunov like et de déterminer ainsi une borne supérieure à la vitesse de convergence. S'il on désire effectuer une étude sur la stabilité uniquement cette inégalité peut être négligée.

La troisième condition représente la contrainte de décroissance de l'énergie sur toutes les régions Ω_q .

Où D_1 est l'ensemble des paires caractérisants les champs de vecteur admissible dans Ω_q . La décroissance de l'énergie doit être inférieur à la valeur négative de la matrice identité afin de conclure à la stabilité exponentielle .On a une stabilité simple si le terme de droite s'annule. Quand à la quatrième condition ,elle représente la contrainte d'une énergie décroissante lors du passage vers une nouvelle région ,ou D_2 représente l'ensemble des régions adjacentes vers lesquelles $x(t)$ peut transiter de Ω_q vers Ω_r .

La stabilité exponentielle est vérifiée s'il existe une solution au LMI problem ci-dessus .les variables α, μ_q , et ζ_q et les matrices P_q sont des inconnues ,contrairement aux différentes matrices Q ,correspondants aux différentes régions générées par l'utilisateur,ou les conditions doivent être vérifiés.

Exemple 3 :

La méthode LMI est illustrée pour le système de l'exemple 3.Afin de démontrer la stabilité exponentielle, l'espace d'état doit être partitionné en deux régions différentes :

- ✦ La première région représente les états ou le premier champ de vecteur est actif c à d V_1 correspond à A_1 .
- ✦ La seconde région représente les états ou le second champ de vecteurs est actif, c à d V_2 correspond à A_2 .

Les deux formes quadratiques $x^T Q_1 x \geq 0$ et $x^T Q_2 x \geq 0$ restreignant les deux fonctions de Lyapunov like à leurs région correspondantes sont données par :

$$Q_1 = \begin{bmatrix} -0.25 & -0.25 \\ -0.25 & 2 \end{bmatrix}, \quad Q_2 = \begin{bmatrix} 0.25 & 0.25 \\ 0.25 & -2 \end{bmatrix}$$

obtenues en utilisant les hyperplans adéquats.

Les trajectoires traversent les frontières des régions dans le sens horaire, impliquant la condition de stabilité : $P_2 \leq P_1$ sur le plan $x_2 = 0.5x_1$ et $P_1 \leq P_2$ sur le plan $x_2 = -0.25x_1$. Qui est formulé dans la troisième condition du LMI problem en formulant ces plans sous formes de formes quadratiques et en utilisant la S-Procédure.

$$P_1 = \begin{bmatrix} 0.1000 & -0.4500 \\ -0.4500 & 41.1167 \end{bmatrix}, \quad P_2 = \begin{bmatrix} 4.3792 & 3.8292 \\ 3.8292 & 6.8833 \end{bmatrix}$$

Avec $\beta = 41.1167$. Alors le système est globalement exponentiellement stable.

Exemple 5 :

La simplification optimale de la dynamique longitudinale d'un avion peut s'écrire sous la forme :

$$\begin{bmatrix} \dot{q} \\ \dot{\alpha} \end{bmatrix} = \dot{x} = f(x, u, p) = \begin{bmatrix} -1 & 10 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} q \\ \alpha \end{bmatrix} + \begin{bmatrix} -1 \\ 0.1 \end{bmatrix} u_{p(t)}.$$

Où $\alpha \leq \alpha_{\text{lim}}$ est la contrainte sur l'angle d'attaque, et q la pente de la vitesse. L'équation de la sortie est sous la forme :

$$\begin{bmatrix} \alpha \\ n_z \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & -300 \end{bmatrix} \begin{bmatrix} q \\ \alpha \end{bmatrix} + \begin{bmatrix} 0 \\ 30 \end{bmatrix} u_{p(t)}$$

Où $p(t) \in \{1, 2\}$, n_z est l'accélération normale et la variable de commande $u_{p(t)}$ est l'angle d'élévation mesuré verticalement à partir de l'horizontale à l'avion.

On a pour ce système un double objectif de commande :

- ✦ Suivre la référence pilote de la normale d'accélération.
- ✦ Maintenir la condition de sécurité $\alpha \leq \alpha_{\text{lim}}$.

Afin de vérifier ces deux objectifs de commande simultanément on se propose la loi du « max control » :

$$p(t^+) = \varphi(x(t), u(t), p(t)) = \arg \max_i (u_i)$$

$$\text{Où} \quad \begin{aligned} u_1 &= -Fx + k_1 \alpha_{\text{lim}} \\ u_2 &= -Gx + k_2 r(t) \end{aligned}$$

Ici, u_1 est la sortie d'un régulateur destiné à stabiliser l'avion autour de α_{lim} . Et u_2 est un régulateur destiné à assurer la poursuite de $r(t)$ par n_z .

D'une manière générale la loi de la « commande maximale » fonctionne tel qu'il y ait poursuite de la fréquence grâce à l'élévateur sauf quand il y a dépassement de l'angle d'attaque. Donc la loi de commande est :

$$u = \max(-Fx + k_1 \alpha_{\text{lim}}, -Gx + k_2 r(t)).$$

Etudions la stabilité de la boucle fermée pour $r(t) \equiv \mathbf{0}$. [53], [61], l'équation du système en boucle fermée devient :

$$\begin{aligned}\dot{x} &= Ax + B \max(-Fx + k_1 \alpha_{\text{lim}}, -Gx) \\ \dot{x} &= (A - BG)x + B \max((G - F)x + k_1 \alpha_{\text{lim}}, \mathbf{0}).\end{aligned}$$

L'étude qui va suivre, suppose que les matrices F et G vérifient : (A-BF) et (A-BG) stables, ce qui est réalisable car la contrôlabilité de (A,B) équivaut à une stabilisation des valeurs propres de A par un retour d'état.

En définissant $\gamma = k_1 \alpha_{\text{lim}}$, et en effectuant un changement de variables, on obtient la forme canonique suivante [53],[61] :

$$\begin{bmatrix} \dot{z}_1 \\ \dot{z}_2 \end{bmatrix} = \begin{bmatrix} z_2 \\ -\alpha_1 z_1 - \alpha_2 z_2 + \max(\alpha_3 z_1 + \alpha_4 z_2 + \gamma, \mathbf{0}) \end{bmatrix}$$

où $\alpha_1, \alpha_2, \alpha_1 - \alpha_3$, et $\alpha_2 - \alpha_4$ sont supérieures à zéro. De plus on suppose que $\gamma \leq \mathbf{0}$, afin que le seul point d'équilibre du système soit l'origine.

La stabilité asymptotique globale est vérifiée si une fonction de Lyapunov appropriée $V(z)$ vérifie les conditions données dans [62].

Jusqu'ici, on a prouvé la stabilité des systèmes en utilisant des fonctions de Lyapunov quadratique. Posons maintenant la fonction :

$$\begin{aligned}V &= \frac{1}{2} \alpha_3 z_1^2 + \int_0^{z_1} [\alpha_1 \xi - \max(\alpha_3 \xi + \gamma, \mathbf{0})] d\xi \\ V &\equiv \frac{1}{2} \alpha_3 x^2 + \int_0^x c(\xi) d\xi\end{aligned}$$

Comme une fonction de Lyapunov candidate pour le système (2.5), étant donné que V est définie positive car $c(\mathbf{0}) = \mathbf{0}$ puisque $\gamma \leq \mathbf{0}$, et $z_1 c(z_1) > \mathbf{0}$ pour $z_1 \neq \mathbf{0}$. Car :

$$\begin{aligned}z_1 c(z_1) &= \alpha x^2 - z_1 \max(\alpha z_1 + \gamma, \mathbf{0}) \\ &= \begin{cases} \alpha_1 z_1^2 > \mathbf{0}, & \alpha_3 z_1 + \gamma \leq \mathbf{0} \\ \alpha_1 z_1^2 - \alpha_3 z_1^2 - \gamma z_1 > \mathbf{0}, & \alpha_3 z_1 + \gamma > \mathbf{0} \end{cases}\end{aligned}$$

, d'où pour le premier cas la condition est vérifiée pour $\alpha_1 > \mathbf{0}$, et pour le second cas on étudie les deux cas $z_1 > \mathbf{0}$ et $z_1 < \mathbf{0}$ séparément qui vérifient eux aussi la condition de positivité.

Afin de vérifier que $\dot{V} \leq \mathbf{0}$, considérons :

$$\begin{aligned}\dot{V} &= \dot{z}_1 \ddot{z}_1 + c(z_1) \dot{z}_1 \\ &= \dot{z}_1 [-\alpha_1 z_1 - \alpha_2 \dot{z}_1 + \max(\alpha_3 z_1 + \alpha_4 \dot{z}_1 + \gamma, \mathbf{0})] + \alpha_1 z_1 \dot{z}_1 - \max(\alpha_3 z_1 + \gamma, \mathbf{0}) \dot{z}_1 \\ &= -\alpha_2 \dot{z}_1^2 + \dot{z}_1 \max(\alpha_3 z_1 + \alpha_4 \dot{z}_1 + \gamma, \mathbf{0}) - \dot{z}_1 \max(\alpha_3 z_1 + \gamma, \mathbf{0}).\end{aligned}$$

Cette dernière inégalité requiert l'étude de 4 cas :

1. si $\alpha_3 z_1 + \alpha_4 \dot{z}_1 + \gamma \leq \mathbf{0}$ et $\alpha_3 z_1 + \gamma \leq \mathbf{0}$ alors $\dot{V} = -\alpha_2 \dot{z}_1^2 \leq \mathbf{0}$.

2. si $\alpha_3 z_1 + \alpha_4 \dot{z}_1 + \gamma > 0$ et $\alpha_3 z_1 + \gamma > 0$ alors $\dot{V} = -(\alpha_2 - \alpha_4) \dot{z}_1^2 \leq 0$.
3. $\alpha_3 z_1 + \alpha_4 \dot{z}_1 + \gamma \leq 0$ et $\alpha_3 z_1 + \gamma > 0$ alors $\dot{V} = -\alpha_2 \dot{z}_1^2 - \dot{z}_1(\alpha_3 z_1 + \gamma)$ est inférieur ou égale à zéro si $\dot{z}_1 \geq 0$ et strictement inférieur à zéro si $\dot{z}_1 < 0$ du moment que $(\alpha_2 - \alpha_4) > 0$
4. $\alpha_3 z_1 + \alpha_4 \dot{z}_1 + \gamma > 0$ et $\alpha_3 z_1 + \gamma \leq 0$ alors $\dot{V} = -\alpha_2 \dot{z}_1^2 + \dot{z}_1(\alpha_3 z_1 + \alpha_4 \dot{z}_1 + \gamma)$ est inférieur ou égale à zéro si $\dot{z}_1 \leq 0$ et strictement inférieur à zéro si $\dot{z}_1 > 0$.

2. Observateurs pour les systèmes dynamiques hybrides :

Le problème de synthèse des observateurs pour les SDH est un sujet qui a commencé très tardivement à attiré l'attention des chercheurs. En effet, la littérature n'est pas très riche sur ce sujet. D'autre part, dans les domaines discrets et continus classiques, la problème de synthèse d'observateur a été largement traitée. Beaucoup de problèmes de base ont été résolus en utilisant différentes approches. La question qui reste souvent posée est comment combiner des techniques déjà établies pour les systèmes discrets et continus pour développer un schéma d'observateur pour les systèmes hybrides ?

On s'est restreint dans ce paragraphe à la synthèse des observateurs pour les systèmes à commutations (SAC) qui sont une classe particulière des SDH, ils ont ouvert des perspectives intéressantes de travaux dans le domaine de la théorie des systèmes de commande. En effet, une large classe de systèmes dynamiques est modélisé par une famille de sous systèmes continus et une règle logique qui gère les commutations entre ces sous systèmes. Plus précisément les SAC consistent en plusieurs sous systèmes qui commutent entre eux selon une loi logique ou d'appartenance; ils peuvent être largement rencontrés en industrie, convertisseurs de puissance, systèmes de contrôle de processus, système de contrôle de température, etc... Pour ces systèmes, il semble raisonnable de concevoir un observateur commuté en synthétisant un ensemble d'observateurs pour chaque mode du SAC. Si le mode actif du SAC est connu, il doit activer seulement le mode correspondant de l'observateur, cependant, il n'y a aucune garantie que l'erreur d'observation converge en utilisant cette approche, même si l'erreur d'observation converge pour chaque mode pris séparément.

2.1 Position du Problème :

Considérons un SDH ayant pour entrée u , sortie y , de composante discrète q et de composante continue x . Soit $\{q_1, \dots, q_N\}$ les ensembles discrets ou location et soit l'évolution de l'état continu x , donnée à travers son équation dynamique dont les paramètres dépendent de l'état discret courant q . La dynamiques du systèmes à commutation peut être donnée par :

$$\dot{x} = f_q(x, u) \quad q \in \{q_1, \dots, q_N\}$$

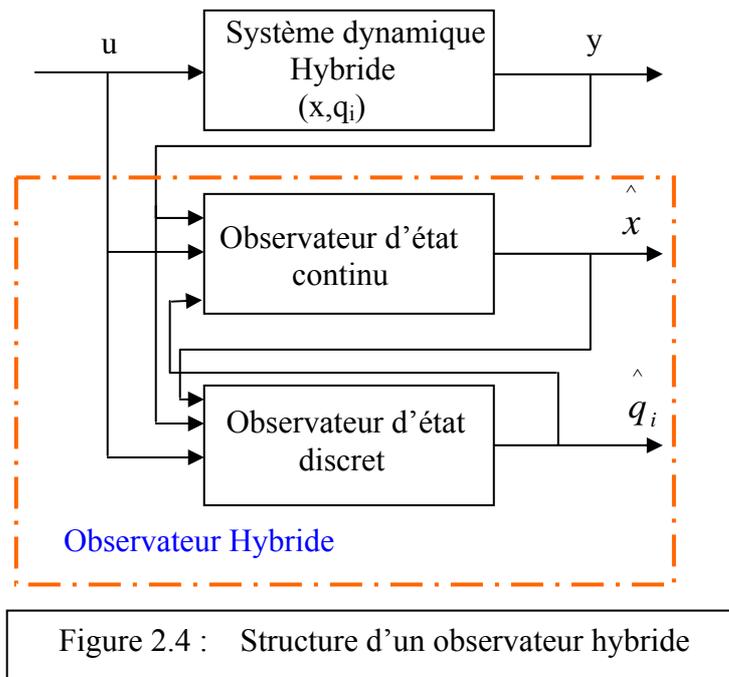
L'objectif de l'observateur hybride est de fournir une estimation de l'état hybride (q, x) à partir de la connaissance de l'évolution de l'entrée u du SDH et de la sortie y . L'observateur considéré est un système hybride avec le même nombre d'états discrets que le système original. Dans chaque location, l'observateur continu doit être synthétisé en utilisant des techniques classiques des observateurs.

La structure de l'observateur hybride illustrée dans la figure 3.1 est composée de deux blocs:

- un observateur d'endroit (temps discret), qui fournit l'estimée \hat{q}_i de la location discrète q_i du système hybride.
- un observateur continu qui fournit l'estimée \hat{x} de l'état continue x du SDH.

La tâche de l'observateur de localisation est de sélectionner la dynamique continue du système dans un ensemble de dynamiques connues. Pour synthétiser un algorithme hybride qui mène à ce résultat, il est nécessaire d'exploiter aussi bien des techniques discrètes que continues.

Le rôle de l'observateur de localisation est d'identifier la dynamique discrète du SDH, en détectant un défaut lorsque le système commute d'un endroit à un autre. Cette tâche peut être suivie d'une logique d'identification pour indiquer l'endroit discret. L'observateur continu a pour rôle d'estimer l'état continu de la dynamique concernée. En fait, à chaque localisation discrète du système hybride correspond une dynamique continue.



2.2. L'interaction entre l'observateur continu et discret :

Nous notons que, en raison du couplage entre l'observateur de l'état discret et l'observateur continu, nous obtenons habituellement l'information discrète seulement après un certain retard en ce qui concerne les instants de commutations [63]. Cependant, nous pouvons réduire ce retard en employant une réinitialisation de l'observateur à chaque signal de commutation par un procédé spécifique. Différentes situations peuvent être considérées. En effet, le couplage entre l'observateur discret et l'observateur continu peut être considéré de deux manières :

- Il est nécessaire de connaître l'état discret q pour considérer l'état continu \hat{x} .
- L'état discret q n'a pas besoin de connaître les estimations des états continus x .

Dans notre cas, le système original est donné sous forme canonique de Fliess, par conséquent, nous sommes dans la deuxième situation. En fait, quand un changement de situation (commutation) se produit, la dynamique continue soudainement change et peut être suivie d'un saut des conditions initiales de l'observateur (et/ou du système lui-même dans d'autres cas). L'identification de ce saut est une des tâches de l'observateur hybride. Dans notre cas, nous ne considérons pas le problème avec saut des conditions initiales quand les commutations se produisent. Malgré cela, le problème de la réinitialisation reste ouvert.

En effet, quand la commutation des dynamiques continues se produisent soudainement, même si l'observateur suit le système parfaitement, à l'instant du saut ou de la commutation soudaine et en raison de sa dynamique, l'observateur continue dans sa direction avant de réaliser le changement de la direction du vrai système. Il essaye alors, aussi rapidement que possible, de dépister encore la vraie trajectoire.

2.3. Observateur par modes glissants :

Un observateur par modes glissants est un observateur dont le terme correcteur est une fonction discontinue. Une des approches utilisées repose sur la théorie des systèmes à structure variable conduisant à la synthèse d'observateurs appelés observateurs par modes glissants.

2.3.1 Observateur par mode glissant d'ordre 1 :

L'idée de l'observateur étape par étape par modes glissants consiste à utiliser le concept du vecteur équivalent ([64] et [65]) de manière itérative.

Considérons le système autonome et non linéaire suivant mis sous forme canonique d'observabilité de Fliess:

$$\begin{cases} \dot{x}_1 = x_2 \dots \dots \dots \dot{x}_{n-1} = x_n \\ \dot{x}_n = f_i(x) \quad i \in \{1, \dots, p\} \\ \text{Si } \sigma_i(x) \text{ est vérifié} \end{cases} \quad (2.8)$$

Où $y = x_1$. et comme le système considéré est donné sous forme canonique d'observabilité, chaque sous système est observable et si on connaît lequel des sous systèmes est actif, on peut alors conclure sur l'observabilité globale, cette classe de systèmes est supposée à état borné en temps fini sans saut et exclut les systèmes Zénon.

REMARQUE 1 : L'hypothèse de la forme canonique du système est quelque peu restrictive. En effet le difféomorphisme associé au système $i = 1$ n'est généralement pas le même que celui associé au système $i = 2$. Ainsi, à chaque commutation, l'état du système mis sous forme canonique d'observabilité de Fliess [64] subit un saut même s'il n'y a pas de saut dans l'état du système d'origine. Néanmoins, pour la majorité des systèmes Lagrangien, cette hypothèse est respectée.

À partir des travaux dans [65], [66] l'observateur à modes glissants suivant a été proposé:

$$\begin{cases} \dot{\hat{x}}_1 = \hat{x}_2 + \lambda_1 \text{sign}(x_1 - \hat{x}_1) \\ \dot{\hat{x}}_2 = \hat{x}_3 + E_1 \lambda_2 \text{sign}(\tilde{x}_2 - \hat{x}_2) \\ \vdots \\ \dot{\hat{x}}_{n-1} = \hat{x}_n + E_{n-2} \lambda_{n-1} \text{sign}(\tilde{x}_{n-1} - \hat{x}_{n-1}) \\ \dot{\hat{x}}_n = f_i(x_1, \tilde{x}_2, \dots, \tilde{x}_n) + E_{n-1} \lambda_n \text{sign}(\tilde{x}_n - \hat{x}_n) \\ \quad i \in \{1, \dots, p\} \quad \text{Si } \sigma_i(x) \text{ est vérifié} \end{cases} \quad (2.9)$$

$$\tilde{x}_j = \hat{x}_j + (\lambda_{j-1} \text{sign}(\tilde{x}_{j-1} - \hat{x}_{j-1}))_{eq}, \quad j = 2 : n$$

Il a été montré que l'erreur d'observation disparaît en temps fini et que les variétés de glissement sont atteintes une à une. Ainsi, une dynamique de dimension 1 est obtenue. Plus précisément $E_i = 0$ s'il existe $j \in \{1, i-1\}$ tel que $\tilde{x}_j - \hat{x}_j \neq 0$ (par définition $\tilde{x}_1 = x_1$), sinon $E_i = 1$ [67].

Dans (2.9) le terme sign_{eq} correspond à l'information d'injection équivalente (par analogie avec le concept bien connu du vecteur équivalent; en d'autre terme la valeur de commande nécessaire pour maintenir un glissement idéal).

REMARQUE 2: L'hypothèse de bornitude de l'état doit concerner tout le système. En effet, les sous systèmes peuvent être parfaitement stable alors que le système global est instable à cause des phénomènes de commutations.

Exemple : Soit le système triangulaire suivant:

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= x_3 \\ \dot{x}_3 &= \begin{cases} P_1(x) & \text{Si } \sigma(x) \geq 1 \\ P_2(x) & \text{Si } |\sigma(x)| < 1 \\ P_3(x) & \text{Si } \sigma(x) \leq -1 \end{cases} \end{aligned}$$

Avec :

$$P_1(x) = -\frac{1800}{49}x_1 - \frac{55}{7}x_2 - \frac{25}{7}x_3 + \frac{2700}{49}$$

$$P_2(x) = \frac{100}{49}x_1 - \frac{36}{7}x_2 - \frac{6}{7}x_3$$

$$P_3(x) = -\frac{1800}{49}x_1 - \frac{55}{7}x_2 - \frac{25}{7}x_3 + \frac{2700}{49}$$

La fonction de commutation : $\sigma(x) = -\frac{7}{100}(x_3 + x_2) - x_1$ avec $y = x_1$

L'observateur associé (2.9) prend la forme suivante :

$$\begin{aligned} \dot{\hat{x}}_1 &= \hat{x}_2 + \lambda_1 \text{sign}(x_1 - \hat{x}_1) \\ \dot{\hat{x}}_2 &= \hat{x}_3 + E_1 \lambda_2 \text{sign}(\tilde{x}_2 - \hat{x}_2) \\ \dot{\hat{x}}_3 &= P_i(x) + E_2 \lambda_3 \text{sign}(\tilde{x}_3 - \hat{x}_3), \quad i \in \{1, 2, 3\} \end{aligned}$$

Avec : $\tilde{x}_2 = \hat{x}_2 + E_1 \lambda_1 \text{sign}(x_1 - \hat{x}_1)$ et $\tilde{x}_3 = \hat{x}_3 + E_2 \lambda_2 \text{sign}(\tilde{x}_2 - \hat{x}_2)$.

Nous allons maintenant appliquer à ce système l'observateur basé sur la technique des modes glissants : modes glissants classique (MG).

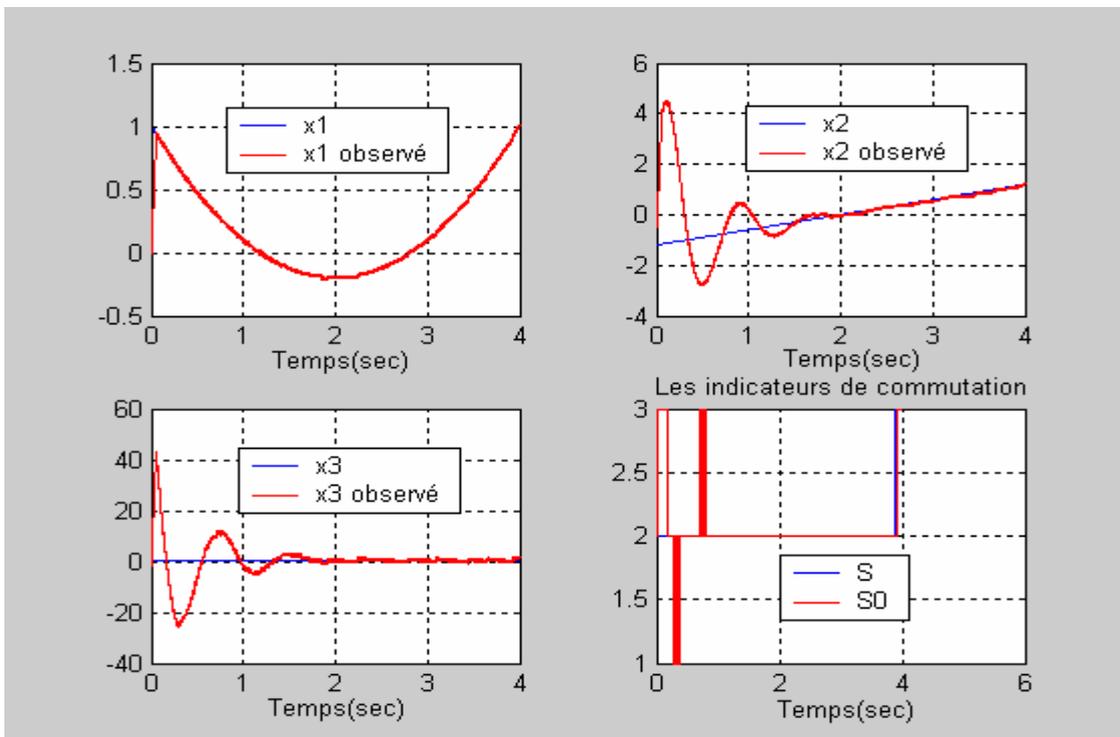


Figure 2.5 : L'application de l'observateur MG sur l'exemple

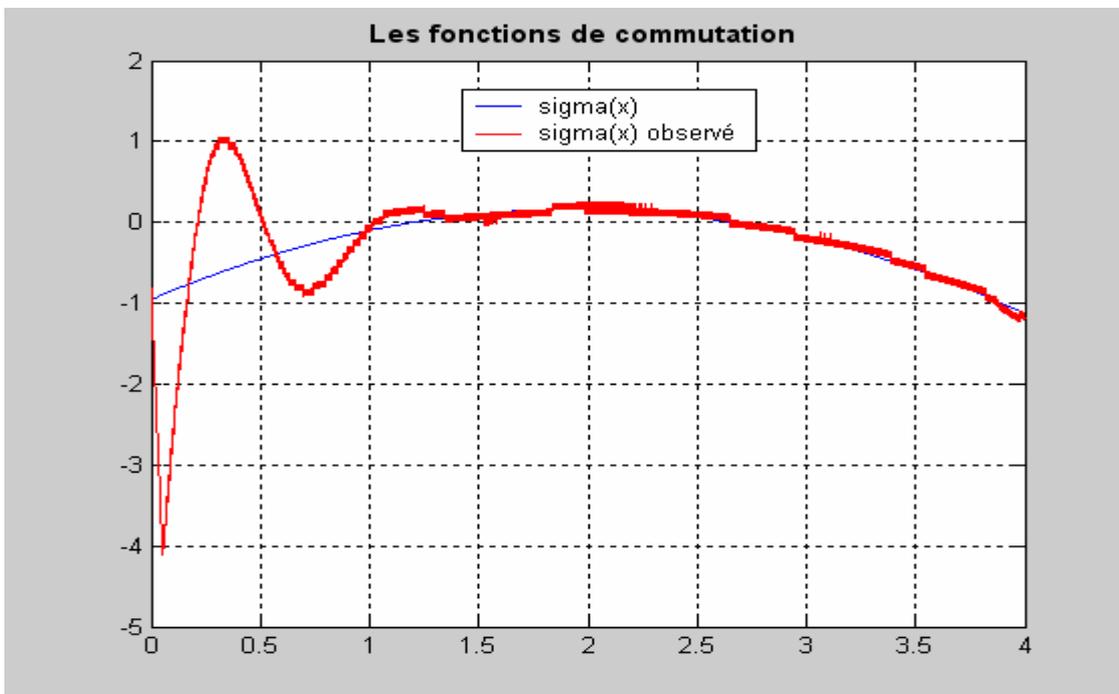


Figure 2.6 : Les fonctions de commutations

La figure 2.5 illustre la convergence en temps fini de l'état du système et les indicateurs de commutations S , et S_0 . (S calculé sur la base de x_1 , x_2 , et x_3 et S_0 calculé sur la base des états observés x_1 , \tilde{x}_2 et \tilde{x}_3). De plus, quand $\sigma(\tilde{x})$ est proche de -1 autour de $t = 4s$, il y a le phénomène de broutement et par conséquent les décisions sont intempestives. La figure 2.6 montre les fonctions de commutations $\sigma(x)$ et $\sigma(\tilde{x})$.

3. Commande optimale des systèmes hybrides :

Contrairement au cas des systèmes continus ou des systèmes à événements discrets pour lesquels la notion de commande et les problèmes de synthèse associés sont bien identifiés et clairement définis, la commande des systèmes hybrides est une notion beaucoup plus large et le domaine de recherche correspondant n'en est qu'à ses débuts. 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 commande hybride rencontrées dans la littérature. Dans cette partie on s'est restreint au problème de commande optimale

3.1. Formulation du problème de commande optimale :

Posons $[t_0 = a, t_1, \dots, t_i, \dots, t_m = b]$ et $[q_0, q_1, \dots, q_i, \dots, q_m]$ (b peut être infini ainsi que m dans ce cas) les suites respectivement des instants de commutation et des modes associés à la commande $(u, d)(.)$ tel que u représente la commande continue et d la commande discrète sur l'intervalle $[a, b]$. On peut alors définir un critère hybride par :

$$J(u, d) = \int_a^b L_{q(t)}(x(t), u(t), t) dt \quad (2.10.a)$$

$$= \sum_{i=0}^m \int_{t_i}^{t_{i+1}} L_{q_i}(x(t), u(t), t) dt \quad (2.10.b)$$

Une commande optimale $(u, d)(.)$ est alors une commande qui minimise J sur l'intervalle $[a, b]$.

La formulation du critère que nous venons de donner n'est pas la plus générale mais l'est suffisamment pour pouvoir illustrer les difficultés liées à la résolution d'un problème de commande optimale hybride. Nous aurions pu entre autre ajouter au critère des coûts associés aux transitions d'état discret et des coûts initial et terminal.

Les apports pratiques actuels se limitent pour l'essentiel aux systèmes commutés.

3.2. La commande optimale sans contraintes des systèmes commutés :

La résolution d'un problème de commande optimale dans le cadre d'un système commuté sans contrainte spatiale ou temporelle, revient à déterminer à la fois une suite optimale comportant les instants de commutation et les modes empruntés et une commande optimale continue associée au mode actif. Ce problème bien que restreint peut être qualifié de difficile.

Pour cette problématique, une approche a également été développée dans ([68], [69], [70]) à l'aide du PM. Nous y faisons l'observation suivante : la dynamique d'un système commuté sans contrainte spatiale ou temporelle peut être décrite par un système continu unique :

$$\dot{x}(t) = \sum_{q=1}^Q \alpha_q(t) f_q(x(t), u(t), t) \in \mathfrak{R}^n \quad (2.11)$$

Avec $\alpha_q(t) \in \{0,1\}$ et $\sum_{q=1}^Q \alpha_q(t) = \mathbf{1}, \forall t$. En effet, dans cette situation comme la fonction de transition discrète se réduit à $q(t^+) \equiv d(t)$ avec un ensemble de commandes discrètes égal à $\underline{D} = \underline{Q}$ on peut parfaitement se ramener à la forme (2.11). A chaque instant, le sous-système actif est donc sélectionné via la commande α_q .

L'application du PM montre alors que la stratégie de commutation doit satisfaire la condition :

A chaque instant t , le quadruplet $(p^*, p_0^*, x^*, q^*)(t)$ vérifie la condition de maximum :

$$H_{q^*}(p^*, p_0^*, x^*, u^*, t) = \max_{q \in \underline{Q}} \sup_{u \in U_q^*} H_q(p^*, p_0^*, x^*, u, t) \quad (2.12)$$

Cette relation signifie que le sous système actif à chaque instant est celui qui possède le plus grand Hamiltonien.

La recherche des instants de commutation est alors déterminée par l'étude des différences $H_{q^*}(p^*, p_0^*, x^*, q^*)(t) - H_{q'}(p^*, p_0^*, x^*, q^*)(t)$ pour la donnée de l'état (x^*, q^*) et de la commande optimale u^* à un instant t .

On retrouve alors pleinement les difficultés liées à la résolution des problèmes optimaux discrets et continus mais ces difficultés sont décuplées par l'imbrication des deux problèmes. Une approche intéressante consiste à discrétiser le problème dès le départ en utilisant la technique de commande prédictive à base de modèle.

3.3. Commande optimale avec contrainte :

Les premiers résultats théoriques et pratiques liés à la commande prédictive ont été obtenus à la fin des années 1970, notamment par les travaux effectués par Richalet *et Al*. Dans les années 1980, plusieurs méthodes basées sur les mêmes concepts prédictifs ont été développées. Parmi celles-ci, citons la Commande Prédictive Généralisée (GPC), développée par David Clarke et son équipe. Ces types de commandes sont aujourd'hui groupés sous la dénomination *model predictive control* «MPC». Dans cette partie, on va introduire brièvement la commande prédictive à base de modèle (MPC) pour les systèmes hybrides linéaires à temps discrets en donnant les principaux traits de cette technique :

- ✦ L'entrée de commande courante, pour la MPC est obtenue en résolvant, à chaque instant d'échantillonnage un problème de commande optimale avec contraintes, en se basant sur les prédictions fournies par un modèle interne du processus commandé.

- ✦ Cette procédure d'optimisation donne une séquence de commande optimale qui minimise une fonction coût donnée ou un critère donné.
- ✦ la MPC combine une commande optimale avec contraintes en boucle ouverte et la technique de receding horizon pour avoir un retour.
- ✦ Les principaux avantages de la technique MPC, incluant sa capacité à gérer systématiquement les contraintes physiques sur les variables à manipuler, états et sorties, ainsi que l'adressage facile de systèmes à plusieurs entrées, plusieurs sorties, ont permis un succès immédiat et une utilisation répandue de la méthode dans le domaine industrielle ces deux dernières décennies.
- ✦ L'un des points les plus importants est que la MPC peut être directement applicable pour les systèmes hybrides sous le formalisme MLD, en prenant ces derniers comme modèle de prédiction [71, 72].

Dans les deux sections suivantes, on va donner une idée générale sur la détermination classique en temps réel de l'entrée de commande et une nouvelle approche de détermination off-line de loi de commande du retour d'état, et cela sans rentrer dans les détails.

3.4. Détermination en temps réel de l'entrée de commande :

Considérons le critère ou la fonction coût suivante :

$$J(x(k), U(k)) = \sum_{l=0}^{N-1} \left\| Q_1(x(k+l|k) - x_{ref}) \right\|_{\{1, \infty\}} + \left\| Q_2(u(k+l|k) - u_{ref}) \right\|_{\{1, \infty\}} \quad (2.12)$$

$$+ \left\| Q_3(y(k+l|k) - y_{ref}) \right\|_{\{1, \infty\}}$$

, qui est contraint par l'effet des matrices Q_i l'évolution prédictive des états, des entrées de commande et des sorties respectivement sur le domaine de prédiction N , en utilisant une norme linéaire (norme 1 ou norme ∞). On suppose que Q_1, Q_2 et Q_3 sont des matrices lignes colonnes pleines.

Le modèle MLD est réinitialisé à chaque pas dans le domaine de prédiction, ce qui permet de construire un vecteur d'optimisation qui contient les séquences des états, entrées, sorties et des variables binaires et réelles auxiliaires, d'où il contient la séquence des entrées de commande $U(k) = [(u(k))^T, \dots, (u(k+N-1))^T]^T$ [73].

Le but étant de trouver la commande adéquate minimisant le critère revient à la résolution d'un problème de commande optimale avec contrainte en temps fini (CFTOC) :

$$U^*(k) = \arg \min_{U(k)} J(x(k), U(k)) \quad (2.13)$$

sous MLD model et $l = k, \dots, k + N - 1$.

L'entrée de commande à l'instant k est obtenue en résolvant le CFTOC (2.13), c'est-à-dire en minimisant le critère (2.12) sous la contrainte imposé par le modèle MLD et les inégalités linéaires mixtes réelles, de plus à la présence d'autres contraintes inégalités sur les variables binaires. Ce qui aboutit à la séquence de commande optimale $U^*(k) = [(u^*(k))^T, \dots, (u^*(k+N-1))^T]^T$. Etant donné qu'on utilise des normes linéaires sur tout le critère alors la résolution du CFTOC revient à une résolution d'un programme mixte linéaire entier (MILP), pour lequel des résultats efficaces ont déjà été prouvés.

3.5. Détermination off line de loi de commande du retour d'état :

En général, le temps nécessaire à la résolution d'un problème de commande optimale en temps réel est supérieur à l'intervalle d'échantillonnage, et le régulateur obtenu non utilisable directement, afin de dépasser ces obstacles, la détermination de loi de commande du retour d'état en « off line » pour tout l'espace d'état est possible en utilisant les outils de programmation multi paramétrique, ou le vecteur d'état est traité comme un paramètre. La loi de commande résultante est une fonction PWA définie par un ensemble de polyèdres réparti sur tout l'espace d'état. Ce type de procédure basée sur une modélisation PWA du système commandé a été récemment introduit pour les systèmes hybrides.

Pour plus de détails sur la programmation multi paramétrique se référer à [74] et annexe E.

Comme il a été proposé dans [75,76], une approche de résolution est la construction de la loi de commande du retour d'état, en combinant mp-LP avec DP (voir Annexe E), Pour cette procédure, le CFTOC doit être reformulé en remplaçant le modèle MLD par son équivalent PWA, ce qui nous donne :

$$U^*(k) = \arg \min_{U(k)} J(x(k), U(k)) \quad (2.14)$$

sous PWA model (2.9) et $l = k, \dots, k + N - 1$.

Une comparaison des deux approches peut être trouvée dans [77]

3.6. Receding Horizon Control :

Le problème CFTOC présenté précédemment donne à l'instant k une séquence d'entrée de commande optimale $U^*(k)$ en boucle ouverte. Afin d'établir un retour, uniquement la première valeur de la séquence $u^*(k)$ est utilisée. A l'intervalle d'échantillonnage suivant, k passe à $k+1$, un nouveau vecteur d'état est obtenu par la mesure ou l'estimation, et le problème de CFTOC est résolu une nouvelle fois sur l'intervalle décalé. Cette technique connue par Receding Horizon Control, est devenue une technique standard dans les récents problèmes de commande. Pour plus de détail se référer à [78].

Conclusion :

Dans ce chapitre on a effectuée une étude générale sur les systèmes hybrides comprenant la stabilité, les observateurs et la commande en présentant les principaux résultats obtenus dans chaque domaine, sans entrer dans les détails. Il nous a permis d'avoir une idée sur les difficultés posées par les systèmes hybrides et les différentes approches utilisées pour les dépassées tout au long de cette étude.

Introduction :

Le système considéré dans cette partie est un banc d'essai incluant deux réservoirs, ce banc d'essai représenté Figure 3.1 est en fait un banc d'essais standard pour des problèmes de commande et de détection de pannes de systèmes hybrides. Le système a été développé au sein du programme de recherche "Control of Complex Systems" (COSY).

3.1. Le système étudié :

Le système hydraulique représenté ci-après se compose de deux réservoirs 1 et 2. Le premier réservoir est alimenté par une pompe de débit Q_1 . Cette pompe est manœuvrée de manière continue de 0 jusqu'à 1. Deux vannes V_1 , V_2 commandent les débits entre les réservoirs. La première vanne 1 est toujours entièrement ouverte, le liquide traverse cette vanne si $h_1 > h_v$. La vanne 2 peut être aussi entièrement ouverte ($V_2=1$), ou entièrement fermée ($V_2=0$). En outre, il y a une fuite permanente du réservoir 2.

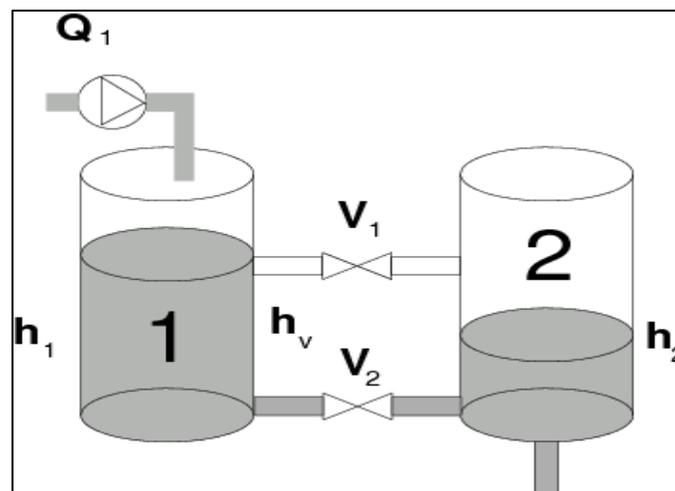


Figure 3.1 : Système de deux réservoirs d'eau

Le système a deux entrées de commande :

- Le débit Q_1 qui peut changer continuellement.
- La position de la vanne 2 qui peut être entièrement ouverte ($V_2=1$), ou entièrement fermée ($V_2=0$). Elle ne permet aucune autre position (vanne TOR).

Le système est soumis à des contraintes sur le débit Q_1 et les niveaux de liquides dans les deux réservoirs. Le système de deux-réservoirs montre des caractéristiques claires des systèmes hybrides. D'abord, la dynamique change selon des conditions de fonctionnement courantes (par exemple il y a un écoulement direct de réservoir 1 au réservoir 2 à travers la vanne 1 si le niveau dans le premier réservoir est au-dessus de certain niveau), et deuxièmement, quelques entrées de système sont commutées entre deux états marche-arrêt, tel que la vanne 2.

3.2. Modélisation :

Pour élaborer le modèle hybride de ce système, on doit étudier toutes les combinaisons possibles de la vanne 2 et le niveau d'eau dans le premier réservoir.

L'équation différentielle décrivant l'évolution du volume dans un réservoir est donné par :

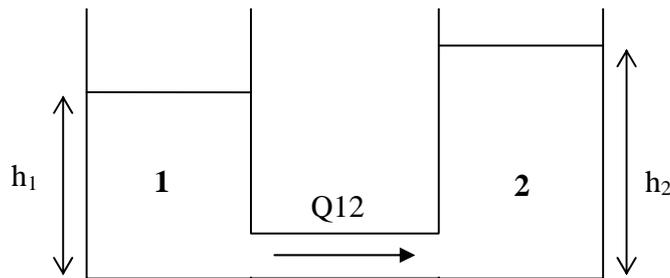
$$\dot{V} = S \dot{h} = \sum Q_{in} - \sum Q_{out}$$

$\sum Q_{in}$: somme des débits entrants

$\sum Q_{out}$: somme des débits sortants

S : la section de réservoir (dans ce cas les deux réservoirs ont la même section).

Sous certaines hypothèses (écoulement laminaire, non compressibilité du liquide, ...) les équations permettent de décrire le comportement dynamique du système (évolution des variables continues) Le débit Q_{12} circulant dans un conduit situé entre les deux réservoirs 1 et 2 (au travers d'une vanne ouverte) est donné par la loi de Toricelli [72]:



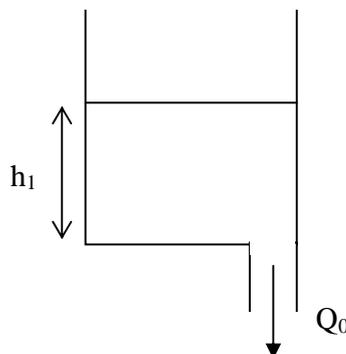
$$Q_{12} = A \operatorname{sign}(h_1 - h_2) \sqrt{2g|h_1 - h_2|} = -Q_{21}$$

Avec A : est la section de la conduite

g : la constante de gravité : $g = 9.81 \text{ m/s}^2$

h_1 et h_2 : hauteur d'eau dans les réservoirs 1 et 2

La loi de Toricelli [72] reste valable dans la configuration suivante :



Dans ce cas on prend $h_2=0$. Ce qui donne :

$$Q_0 = A\sqrt{2g h_1}$$

En utilisant les résultats précédents, le modèle de notre système est donné par :

Cas 1 : $h_1 \geq h_v$ et $V_2 = 1$

La conservation de la masse dans les réservoirs fournit les équations différentielles :

$$\begin{cases} S\dot{h}_1 = Q_1 - A\sqrt{2g(h_1 - h_v)} - A\sqrt{2g(h_1 - h_2)} \\ S\dot{h}_2 = A\sqrt{2g(h_1 - h_v)} + A\sqrt{2g(h_1 - h_2)} - A\sqrt{2gh_2} \end{cases}$$

Cas 2 : $h_1 < h_v$ et $V_2 = 1$

Dans ce cas le modèle est le suivant :

$$\begin{cases} S\dot{h}_1 = Q_1 - A\sqrt{2g(h_1 - h_2)} \\ S\dot{h}_2 = A\sqrt{2g(h_1 - h_2)} - A\sqrt{2gh_2} \end{cases}$$

Cas 3 : $h_1 \geq h_v$ et $V_2 = 0$

$$\begin{cases} S\dot{h}_1 = Q_1 - A\sqrt{2g(h_1 - h_v)} \\ S\dot{h}_2 = A\sqrt{2g(h_1 - h_v)} - A\sqrt{2gh_2} \end{cases}$$

Cas 4 : $h_1 < h_v$ et $V_2 = 0$

$$\begin{cases} S\dot{h}_1 = Q_1 \\ S\dot{h}_2 = -A\sqrt{2gh_2} \end{cases}$$

On aura ainsi le modèle hybride du système avec quatre modes de fonctionnement indexé par 1, 2, 3 et 4 respectivement.

Le but est de concevoir un contrôleur qui peut stabiliser le niveau de liquide dans le deuxième réservoir au point de référence $h_{2ref} = 0.2$ m. Le contrôleur peut manoeuvrer le débit Q_1 continuellement dans l'intervalle $[0,1]$ et la position de la vanne 2, qui peut seulement prendre les valeurs booléennes 0 ou 1. Le contrôleur doit faire face aux différentes contraintes et avoir les performances désirées.

Pour pouvoir modéliser le système sous la forme MLD, il convient d'approximer les termes non linéaires donnant les débits par les équations linéarisées suivantes :

$$\begin{cases} \sqrt{2g(h_1 - h_v)} = \sqrt{\frac{2g}{h_{\max} - h_v}} (h_1 - h_v) \\ \sqrt{2g(h_1 - h_2)} = \sqrt{\frac{2g}{h_{\max}}} (h_1 - h_2) \\ \sqrt{2gh_2} = \sqrt{\frac{2g}{h_{\max}}} h_2 \end{cases}$$

Tel que h_{\max} est le niveau d'eau maximal dans le premier réservoir et h_v est la hauteur de la première vanne. Ces expressions avec l'approximation d'Euler de premier ordre ($\dot{h} \rightarrow \frac{h(k+1) - h(k)}{T_s}$) permettent alors l'élaboration d'un modèle MLD.

Un modèle MLD peut être obtenu par les techniques classiques exposées par Bemporad *et al* en introduisant les variables auxiliaires continues et binaires requises par la traduction des interfaces. On fera intervenir les vecteurs suivants :

$$\begin{aligned} x &= [h_1 \quad h_2]^T \\ u &= [Q_1 \quad V_2]^T \end{aligned}$$

Les paramètres du modèle sont donnés dans le tableau ci-dessous :

PARAMETRE	VALEUR
A	10^{-6} m^2
h_v	0.3 m
S	0.0143 m^2
h_{\max}	0.65 m
La période d'échantillonnage	$T_s = 10 \text{ s}$

Tableau 3.1 : Les paramètres du modèle

La stratégie de commande développée lors du paragraphe précédent est appliquée pour la régulation de niveau. Les pondérations de la fonction coût sont données par :

$$Q_1 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, Q_2 = \begin{bmatrix} 10^{-5} & 0 \\ 0 & 10^{-5} \end{bmatrix} \text{ et } Q_3 = 200$$

pour un horizon de prédiction $N=3$.

Une fois le modèle et le problème sont posés, le contrôleur peut être obtenu par un outil de résolution. L'outil le plus complexe et probablement le plus performant est la bibliothèque MPT, développé à ETH Zurich. Il intègre des modules performants d'optimisation (YALMIP) permettant la construction des solutions explicites.

Le comportement du système en boucle ouverte est représenté ci-dessous :

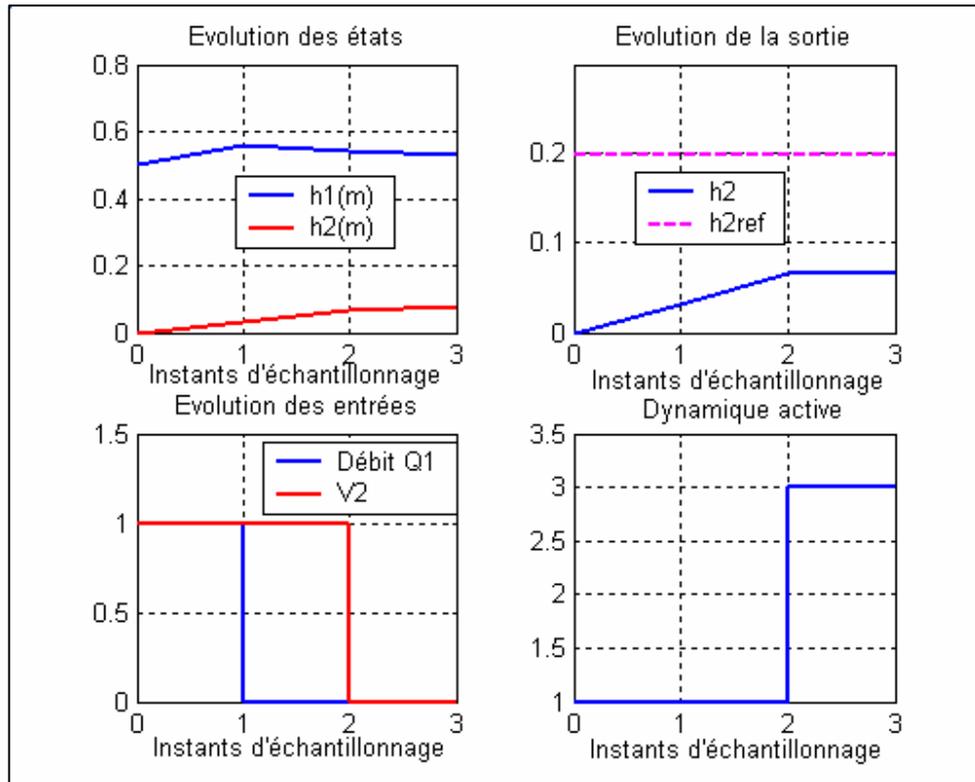


Figure 3.2 : Le comportement du système en boucle ouverte

D'après le théorème, le régulateur obtenu est une partition de polyèdres comme illustre la figure 3.3 .

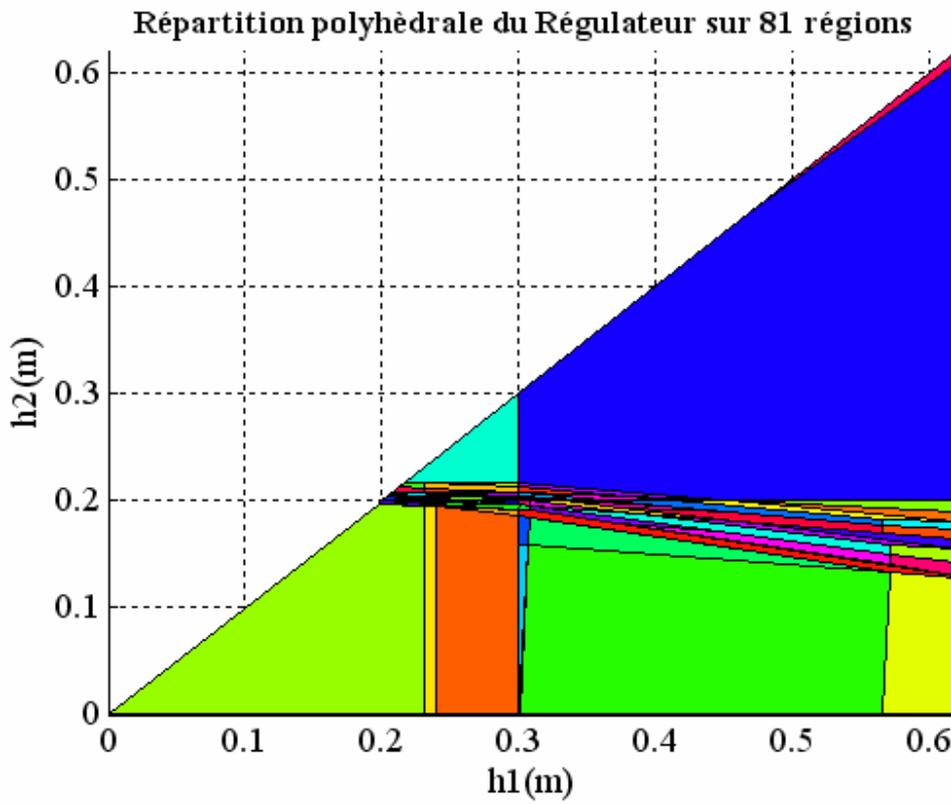


Figure 3.3 : La loi MPC pour la régulation de niveau d'eau

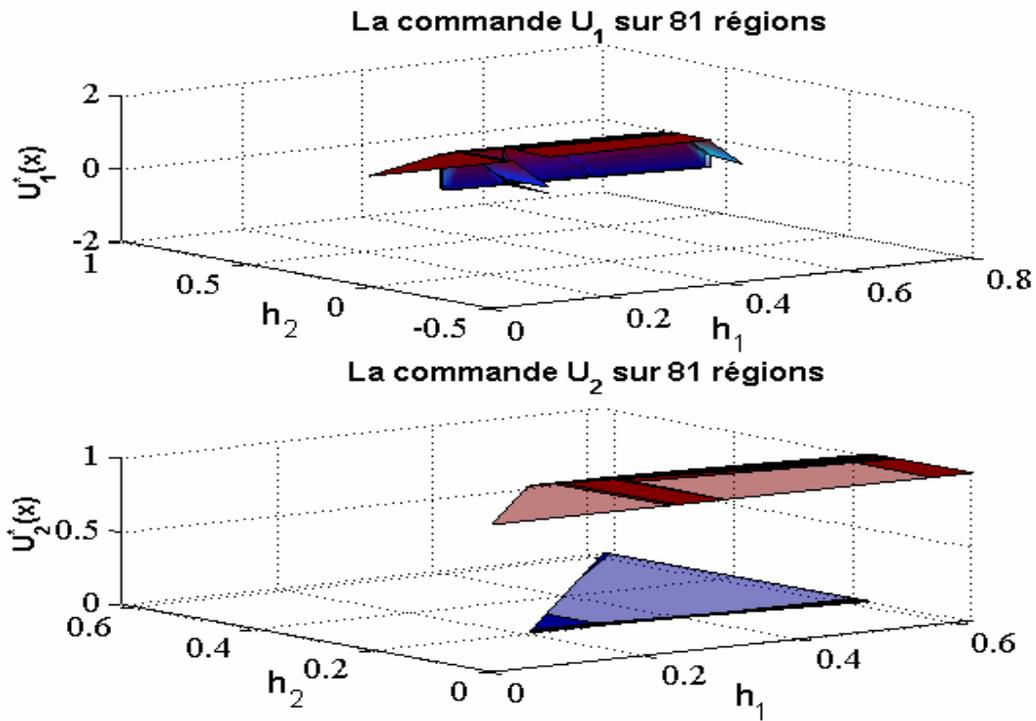


Figure 3.4 : La commande optimale en fonction des niveaux d'eau

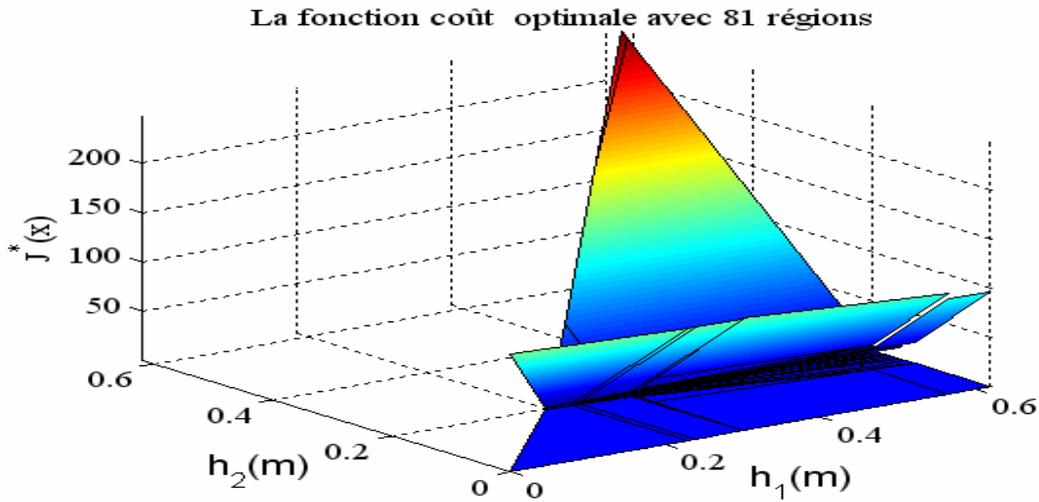


Figure 3.5 : La valeur de la fonction optimale

La simulation en boucle fermée est donnée par la figure ci-dessous à partir des conditions initiales $h_1=0.5\text{m}$ et $h_2=0\text{ m}$.

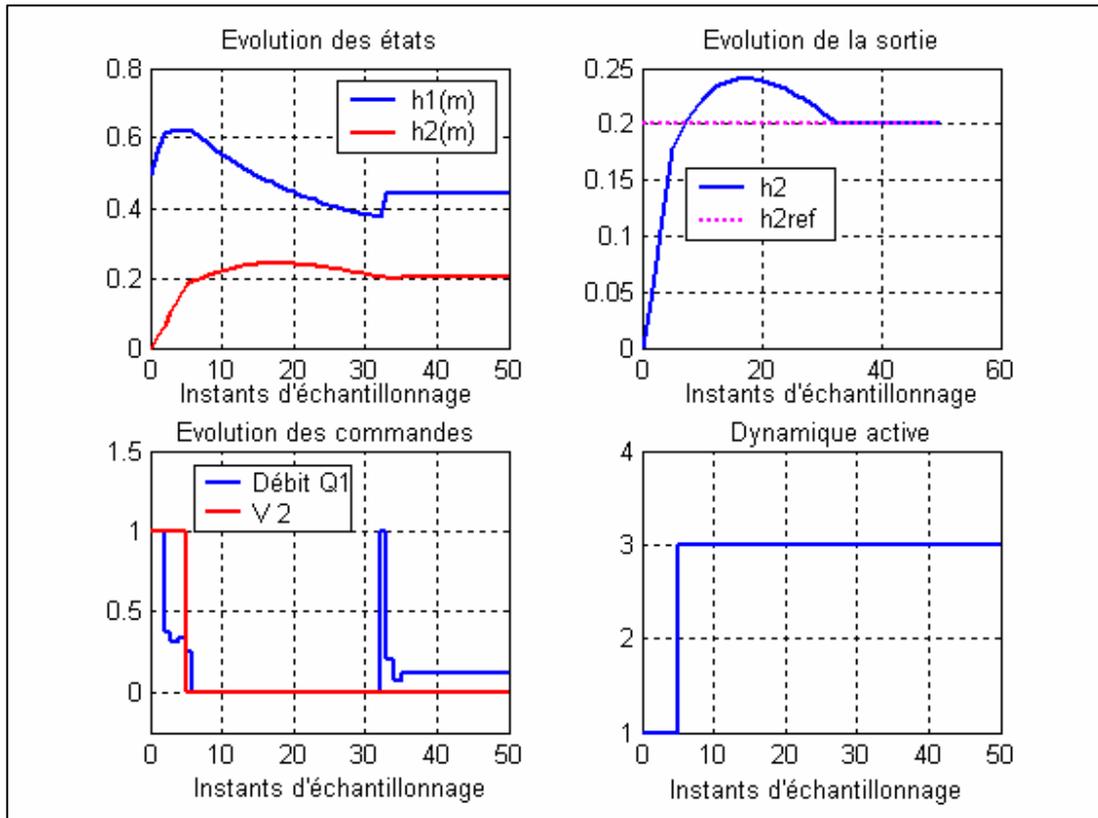


Figure 3.6 : Simulation de comportement du système bouclé avec la loi MPC

Les résultats de simulation illustrent les performances de la loi MPC, le niveau d'eau dans le réservoir 2 se stabilise à la valeur de référence au bout de 3sec avec un dépassement acceptable.

Le contrôleur peut être encore simplifié au nombre réduit de régions. La simplification est basée sur le fusionnement des régions qui ont la même loi de commande, qui garantit que le contrôleur simplifié maintiendra les mêmes performances que le contrôleur original. Le contrôleur simplifié se compose de 51 régions représenté par la figure suivante.

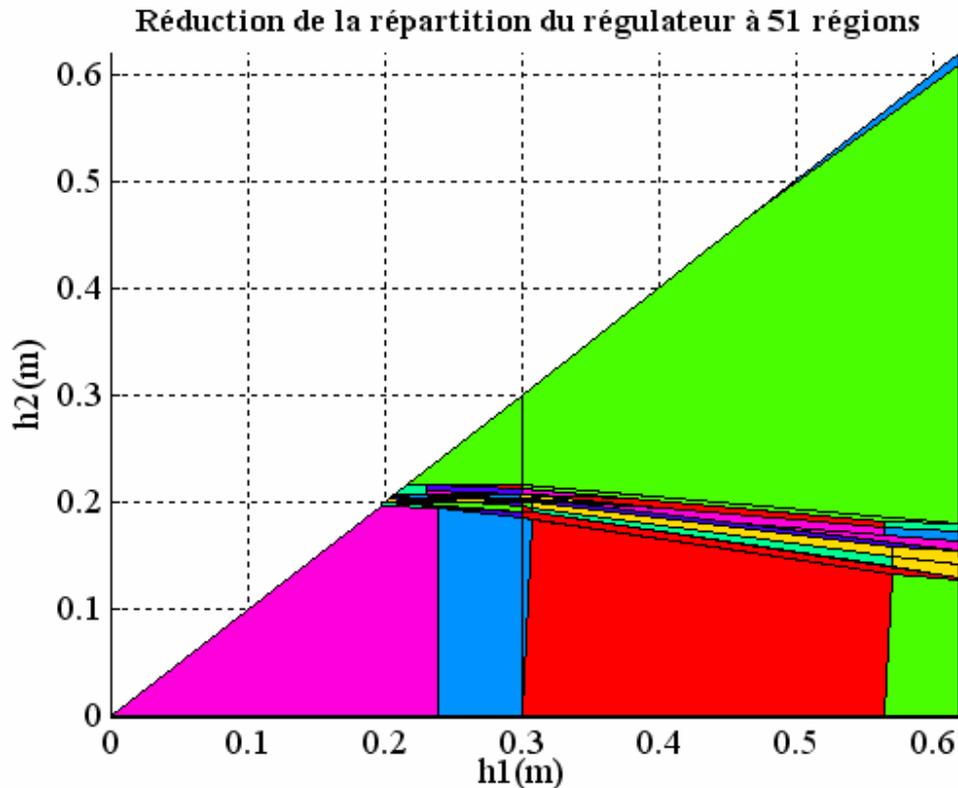


Figure 3.7 : Le régulateur MPC de niveau simplifié

Conclusion :

Dans cette partie on a abordé le problème de régulation du niveau d'eau d'un réservoir en se basant sur la technique de commande prédictive à base de modèle. Après avoir établi le modèle du système et posé les paramètres du critère à optimiser, le régulateur est synthétisé par la bibliothèque MPT, ce contrôleur répond aux performances demandées.

Introduction :

Le convertisseur à commutation de modèle DC-DC est un montage de l'électronique de puissance, utilisé grâce à son faible poids, sa taille réduite, son efficacité et sa grande précision dans une large variété d'applications. Il constitue la technologie de base dans les différentes alimentations, dans les ordinateurs, les chargeurs, dans des applications médicales et aérospatiales délicates ou précises, ainsi que dans la commande des moteurs à courant continu et vitesse variable [79]. C'est pourquoi leurs analyses, études et modélisations, en boucle ouverte ou fermée ont attiré un large spectre de chercheurs, et d'industrielles en quête à des techniques de commandes révolutionnaires et efficaces.

Etant donné une tension en entrée continue non commandable, telle que la sortie d'un filtrage redressement d'une tension alternative, et une tension de sortie nécessitant d'importantes variations dans le temps constituant ainsi une charge variable; il fallait introduire une commande permettant une régulation de la tension de sortie, en présence des perturbations causées par la tension d'entrée et la charge variable.

Les difficultés rencontrées lors de la commande du convertisseur DC-DC, reviennent à l'aspect intrinsèque hybride de ce dernier; D'une manière générale, ce convertisseur est constitué de trois modes de fonctionnement différents, ou à chacun est associée une dynamique linéaire continue dans le temps, à cette difficulté s'ajoute la présence de contraintes résultantes de la topologie de ce hacheur, particulièrement la délimitation physique de la variable de commande représentée par le rapport cyclique entre zéro et un et la contrainte de positivité de la première variable d'état, le courant d'induction, durant le mode discontinu; aux quelles s'ajoutent d'autres contraintes résultantes des conditions de sécurité, tel que le courant limite ou un démarrage adouci qui agissent directement sur les valeurs du courant. Le problème de commande est encore plus compliqué par d'importantes variations du point de fonctionnement, dues aux perturbations cités précédemment.

Le convertisseur DC-DC est un circuit à commutation qui transfère une tension d'une source continue, vers une charge, à l'aide d'un interrupteur à semi-conducteur qui est périodiquement mis sur on ou off et d'un étage d'un filtre passe bas, avec une capacité et une inductance, ce qui donne une tension de sortie continue à faible ondulation; L'interrupteur est commandé par une séquence d'impulsion périodique avec une fréquence constante appelée fréquence de commutation f_s (période de commutation T_s), caractérisant le fonctionnement du convertisseur. La composante à courant continu de la tension de sortie est régulée à l'aide du rapport cyclique $d = t_{on} / T$, ou t_{on} représente l'intervalle de conduction du convertisseur.

L'approche de modélisation la plus utilisée pour ce convertisseur est celle basée sur la moyenne de l'espace d'état ou « state-space averaging » [80,81]; dans le but de dépasser les difficultés dues à l'aspect hybride du système, un modèle moyen ou approximatif à temps continu prenant le rapport cyclique comme entrée, et décrivant les dynamiques lentes du système est mis au point. Ce dernier reste néanmoins non linéaire à cause de la présence de multiplication entre la variable de commande et les variables d'état. Quand à la conception du régulateur elle est obtenue en utilisant les techniques de commande linéaire sur un modèle linéarisé autour d'un point de fonctionnement. L'un des aspects de l'insuffisance de cette technique d'approximation est l'omission des dynamiques rapides du système lors de la modélisation, ainsi que des oscillations sous harmoniques.

Une approche plus rigoureuse est la description du système à l'aide de modèles discrets dans le temps, qui dressent un plan ou décrivent les variable d'état, du début à la fin de la période de commutation [82,83].

Cette méthode décrit avec succès plusieurs aspects complexe de la dynamique du convertisseur DC-DC et elle est très utile pour l'étude de phénomènes telles que les oscillations sous harmoniques et aléatoires observées lors du fonctionnement en boucle fermée [84]. Néanmoins pour des fins de modélisation ou de commande, le problème de la non linéarité reste posé. Par conséquent il n'existe toujours pas d'approche systématique pour la conception du régulateur adéquat.

Le principal objectif de commande du convertisseur DC-DC est la commutation de l'interrupteur à semi-conducteur avec un rapport cyclique permettant de confondre la composante continue de sa tension de sortie, avec sa référence, et cela malgré les variations de la tension d'entrée et de la charge. Le concept de base utilisé actuellement pour la commande des convertisseurs DC-DC, est la modulation de la largeur d'impulsion « MLI », où l'interrupteur est mis à un au début de chaque période de commutation, puis mis à zéro par le régulateur lors de la vérification de certaines conditions. Un verrou maintient l'interrupteur à zéro jusqu'au début de la période suivante. Avec cette configuration le problème de commande est de déterminer l'instant de mise à zéro de l'interrupteur.

Dans la littérature, un grand nombre de stratégies ont été introduite pour la conception d'un régulateur adéquat. Ces dernières varient de la logique flou [85] à la régulation linéaire quadratique « LQR » [86,87,88]; des méthodes de commande non linéaire [89,90,91] à la régulation feed forward [92,93]. Le point commun à ces différentes approches est l'utilisation de modèles simplifiés pour la modélisation du convertisseur. Il est clair que l'utilisation de ce type de modèles localement ou moyennement linéarisable, ne permet pas de capturer toutes les caractéristique de la dynamique complexe dues à l'aspect hybride du hacheur DC-DC. et limite forcément l'espace atteignable, donnant ainsi des résultats limités ou conditionnés. Particulièrement pour la commande LQR dans [94,95] ou des modèles à temps discrets linéarisés autour d'un point de fonctionnement sont utilisés, et pour la commande non linéaire dans [96,97,98] ou l'aspect hybride est surmonté par une modélisation moyenne lors de la conception du régulateur. De plus aucune des modélisations précédentes ne prend en compte les contraintes.

Des travaux récents sur la nature hybride des convertisseurs DC-DC tendent à utiliser cette caractéristique pour la conception d'un régulateur robuste [99,100].

Motivées par de telles difficultés, on présente dans ce chapitre une nouvelle approche de modélisation et de conception de régulateurs pour les convertisseurs DC-DC établie par Manfred Morarri et ses collaborateurs en 2003.

Le système sur lequel l'étude a été effectuée est un convertisseur DC-DC abaisseur. Il a été modélisé comme un système dynamique hybride sous le formalisme MLD, en introduisant la notion de ν -step model, ce qui a donné un modèle valide sur tous les modes de fonctionnement, et décrivant l'évolution des variables d'état sur une période entière. En se basant sur le modèle MLD, on formule et résout un problème de commande optimale en temps fini. Le résultat est un régulateur vérifiant tout les objectifs de commande pour un point de fonctionnement quelconque.

3.1. Modélisation du convertisseur :

On débute par une modélisation du convertisseur dans le domaine des temps continus, afin de tirer les équations correspondantes à chaque mode de fonctionnement. Ce modèle prend en compte les perturbations, plus particulièrement, la résistance interne de la self et la résistance série équivalente de la capacité (RSE).

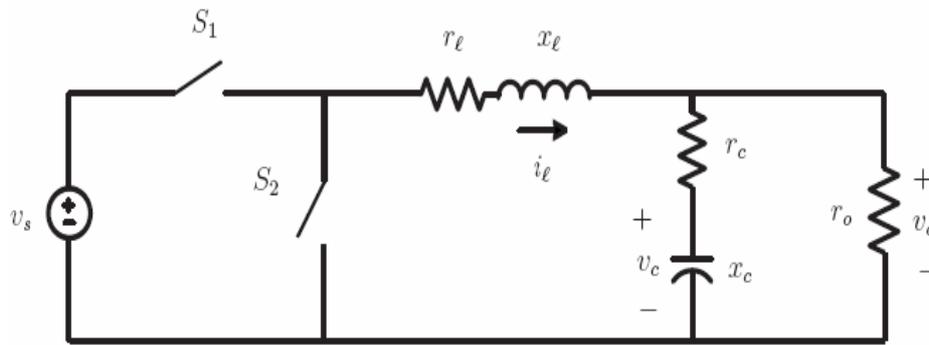


Figure 3.8 : Topologie du convertisseur abaisseur DC-DC

3.1.1. Le modèle à temps continu :

La topologie du circuit du convertisseur DC-DC est donnée à la figure 1. Les quantités utilisées étant toutes normalisées, représentent :

v_s La tension d'entrée, r_l la résistance interne de l'inductance, x_l la self, r_c la RSE de la capacité, x_c la capacité, r_o et v_o la résistance de charge et la tension de sortie respectivement .

- ✦ Pour chaque période k , un rapport cyclique $d(k)$ compris entre zéro et un est déterminé par le régulateur .
- ✦ Durant l'intervalle $kT_s \leq t < (k + d(k))T_s$, l'interrupteur S_1 conduit, et la tension est transférée directement de la source vers la charge. Durant la conduction de S_1 , S_2 est ouvert .A la fin de cet intervalle S_1 est ouvert et maintenue ainsi jusqu'au début de la prochaine période.
- ✦ L'interrupteur S_2 qui fonctionne en opposition avec S_1 est lui fermé pour $(k + d(k))T_s \leq t < (k + 1)T_s$, moment durant lequel S_2 fournit un passage au courant d'induction i_l , indifféremment de son signe .

b. Modèle du circuit :

On pose $x(t) = [i_l(t) \quad v_c(t)]^T$, comme le vecteur d'état, où $i_l(t)$ est le courant d'induction et $v_c(t)$ la tension aux bornes de la capacité, et soit $d(k)$ le rapport cyclique au cours de la k me période .Le système est décrit par l'ensemble des équations d'états affines et continues suivantes :

S_1 conduit :

$$\dot{x}(t) = Fx(t) + fv_s, \quad kT_s \leq t < (k + d(k))T_s \quad (3.1)$$

S_1 ouvert :

Le système évolue en autonomie c à d :

$$\dot{x}(t) = Fx(t), \quad (k + d(k))T_s \leq t < (k + 1)T_s \quad (3.2)$$

Où les matrices F et f sont données par :

$$F = \begin{bmatrix} -\frac{1}{x_l} \left(r_l + \frac{r_o r_c}{r_o + r_c} \right) & -\frac{1}{x_l} \frac{r_o}{r_o + r_c} \\ \frac{1}{x_c} \frac{r_o}{r_o + r_c} & -\frac{1}{x_c} \frac{1}{r_o + r_c} \end{bmatrix} \quad (3.3)$$

$$f = \begin{bmatrix} \frac{1}{x_l} \\ \mathbf{0} \end{bmatrix}$$

La tension de sortie au borne de la charge est exprimée comme une fonction d'état tel que :

$$v_o(t) = g^T x(t) \quad (3.4)$$

Avec :

$$g = \begin{bmatrix} \frac{r_o r_c}{r_o + r_c} & \frac{r_o}{r_o + r_c} \end{bmatrix}^T \quad (3.5)$$

La variable qui nous intéresse le plus d'un point de vue commande est l'erreur sur la tension de sortie, qui est obtenue par l'intégration de la différence entre la tension de sortie et sa référence le long d'une période de commutation k :

$$v_{o,err}(k) = \int_{kT_s}^{(k+1)T_s} (v_o(t) - v_{o,ref}) dt \quad (3.6)$$

où $v_{o,ref}$ dénote la référence de la tension de sortie.

En résumé le convertisseur se caractérise par deux modes de fonctionnement, avec deux dynamiques différentes. Ces deux modes dont la fonction de sortie est identique, se différencient seulement dans les expressions de leurs équations d'états affines.

Au début de chaque période, le mode 1 généré par (3.1) est actif, puis le rapport cyclique $d(k)$ détermine le moment de commutation du premier au second mode qui évolue suivant (3.2).

Il est important de noter qu'en pratique, le courant d'induction $i_l(t)$ et la tension de sortie $v_o(t)$ sont directement mesurables, ainsi que les variations de la tension d'entrée $v_s(t)$ d'après les travaux effectués dans [101], ce qui permet de tirer aisément la tension aux bornes de la capacité $v_c(t)$. Comme il est possible en considérant la RSE de la capacité très petite (négligeable) de supposer que $v_c(t) = v_o(t)$ aux instants d'échantillonnage k , ce qui introduit une très faible erreur sans grand effet.

Les contraintes présentent dans notre modèle proviennent de deux sources différentes :

- ✦ Par définition, le rapport cyclique $d(k)$ est contraint à être compris entre zéro et un .
- ✦ Le fait que les composants à semi-conducteurs ainsi que les charges ne peuvent supporter réellement qu'une certaines valeurs maximale de courant implique une contrainte sur le courant d'induction qui est borné par une valeur de courant maximale variant suivant l'application tel que : $i_l(t) \leq i_{l,max}(t)$.

Les paramètres du convertisseurs DC-DC sont généralement variables dans le temps ,néanmoins cette variation peut être répartie en deux catégories ,la première inclue les paramètres de l'étage du filtre passe bas x_c et x_l qui ne sont sujet qu'à des détériorations très lentes par l'effet de la chaleur ou de la température de même que la RSE r_c de la capacité et la résistance interne de la bobine r_l ,qui sont considéré dans notre cas comme des quantités invariantes dans le temps. Par contre dans la seconde catégorie on a la tension d'entrée et la résistance de charge qui elles peuvent varier d'une manière très importante. De plus aux courant limite et à la tension de référence qui n'ont pas été restreint à être constants par le cahier de charge.

a. Le modèle de prédiction :

Dans cette partie on va reformuler le modèle ci-dessus, afin de pouvoir appliquer la commande optimale.

Premièrement, d'un point de vue programmation, il est préférable que toutes les variables d'état soient mesurables directement, c'est pourquoi on remplace dans le vecteur d'état, la tension aux bornes de la capacité, par la tension de sortie. Ce qui conduit au nouveau vecteur d'état

$\mathbf{x}(t) = \begin{bmatrix} i_l(t) & v_o(t) \end{bmatrix}$,et les matrices F,f et g deviennent :

$$F = \begin{bmatrix} -\frac{r_l}{x_l} & -\frac{1}{x_l} \\ \frac{1}{x_c} \frac{r_o}{r_o + r_c} (1 - x_c r_c \frac{r_l}{x_l}) - \frac{1}{x_c} \frac{1}{r_o + r_c} (1 + x_c r_c \frac{r_o}{x_l}) \end{bmatrix}, \quad (3.7)$$

$$f = \begin{bmatrix} \frac{1}{x_l} \\ \frac{r_o}{r_o + r_c} \frac{r_c}{x_l} \end{bmatrix}, \quad g = [0 \quad 1]^T$$

Deuxièmement, il faut que la tension d'entrée v_s soit un paramètre de la loi de commande finale, afin de pouvoir agir directement sur les variations de cette dernière. C'est pourquoi elle a été éliminé du modèle obtenu précédemment en normalisant toutes les quantités physiques par rapport à elle, et en introduisant un vecteur d'état normalisé $\mathbf{x}'(t) = \frac{\mathbf{x}(t)}{v_s}$ les équations (3.1),(3.2),(3.4) et (3.6)

deviennent :

$$\frac{d\mathbf{x}'(t)}{dt} = \begin{cases} F\mathbf{x}'(t) + f, & kT_s \leq t \leq (k + d(k))T_s. \\ F\mathbf{x}'(t), & (k + d(k))T_s \leq t \leq (k + 1)T_s \end{cases} \quad (3.8.a)$$

$$v_o'(t) = g^T \mathbf{x}'(t),$$

Ou F et f sont données dan (3.7) et $v_o' = \frac{v_o}{v_s}$ est la tension de sortie normalisée.

La formule de l'erreur sur la tension de sortie devient :

$$v'_{o, err}(k) = \int_{kT_s}^{(k+1)T_s} (v'_o(t) - v'_{o, ref}) dt \quad (3.9)$$

avec la référence sur la sortie normalisée : $v'_{o, ref} = \frac{v_{o, ref}}{v_s}$, et l'erreur normalisée $v'_{o, err} = \frac{v_{o, err}}{v_s}$.

De même le courant pour le courant limite :

$$i'_{l, max} = \frac{i_{l, max}}{v_s} \quad (3.10)$$

Comme précédemment, les matrices F et f sont supposées invariantes dans le temps. On ajoute l'hypothèse que la résistance de charge est constante. Grâce à la mise à l'échelle, les équations du modèle sont devenues indépendantes de la variable v_s , et les seuls paramètres variables dans le temps du modèle sont la tension mise à l'échelle de référence $v'_{o, ref}$ et le courant max mis à l'échelle $i'_{l, max}$.

En résumé le modèle donné par les équations (3.7)-(3.10) contient deux variables d'état normalisées et deux paramètres. Ce modèle sera utilisé par la suite pour le problème de commande optimale avec contraintes.

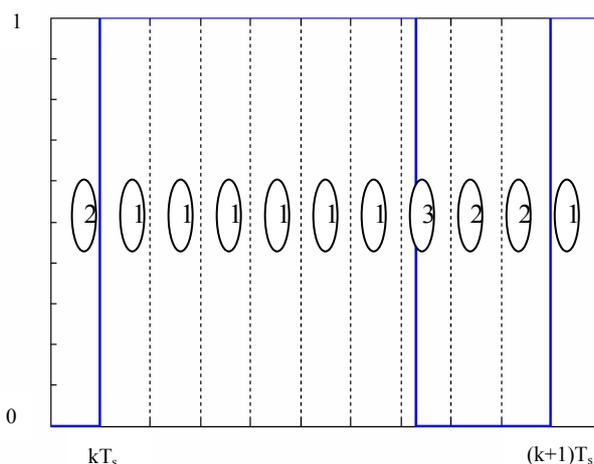


Figure 3.9 : Partition d'une période de commutation suivant le ν -step model

3.1.2. Le modèle hybride discrétisé « ν -step model » :

Le but de cette section est de tirer un modèle adéquat au problème de commande optimale, en se basant sur le modèle de prédiction posé précédemment, ce modèle doit vérifier les propriétés suivantes :

1. Il est clair que le régulateur et le modèle doivent être exprimés dans le domaine des temps discrets, étant donné une variable de commande représentée par le rapport cyclique $d(k)$ ne variant qu'aux instants d'échantillonnage kT_s .
2. Il est préférable de décrire l'évolution des états sur toute la période et non seulement aux instants d'échantillonnage, ce qui permet de garder les pics de variation du courant d'induction $i_l(t)$ en dessous de la valeur limite à tout moment.
3. Le modèle doit fournir une approximation de l'erreur sur la tension de sortie donnée par l'intégrale (3.9).

Le plus important est de pouvoir maintenir ou respecter la nature hybride intrinsèque du convertisseur en gardant les deux modes de fonctionnement.

Basée sur de telles considérations, la notion du ν -step model a été introduite, elle vérifie tout les points cités ci-dessus en divisant la période de longueur T_s en ν sous périodes de longueur $\tau_s = \frac{T_s}{\nu}$, avec $\nu \in \mathbb{N} \geq 2$. Comme le décrit la figure 2. On note les états d'une sous période d'échantillonnage τ_s par $\boldsymbol{\varepsilon}(n)$, où n renvoie aux instants discrets d'une sous période d'échantillonnage $n = \{1, \dots, \nu - 1\}$

On a par définition :

$$\begin{cases} \boldsymbol{\varepsilon}(0) = \boldsymbol{x}'(k). \\ \boldsymbol{\varepsilon}(\nu - 1) = \boldsymbol{x}'(k + 1) \end{cases}$$

Notez, que $\boldsymbol{\varepsilon}$ renvoie au vecteur normalisé \boldsymbol{x}' .

Introduisant maintenant ν variables binaires tel que :

$$\sigma_n = \text{vrai} \Leftrightarrow d(k) \geq \frac{n}{\nu}; \quad n = 0, \dots, \nu - 1 \quad (3.11)$$

Qui représentent les positions échantillonnées de l'interrupteur S_1 aux instant $n\tau_s$. Sans oublier que S_2 est en opposition avec S_1 .

On introduit pour chaque sous période, les deux modes de fonctionnement présentés précédemment, plus un troisième mode auxiliaire qui représente la transition du mode 1 vers le mode 2 :

- **Mode 1** : l'interrupteur S_1 est fermé pendant toute une sous période.
- **Mode 2** : l'interrupteur S_1 est ouvert pendant toute une sous période.
- **Mode 3** : l'interrupteur S_1 s'ouvre durant une sous période.

D'où pour la $n^{\text{ème}}$ sous périodes on a les équations :

$$\boldsymbol{\varepsilon}(n + 1) = \begin{cases} \Phi \boldsymbol{\varepsilon}(n) + \Psi, & \sigma_n \wedge \sigma_{n+1}. \\ \Phi \boldsymbol{\varepsilon}(n), & \bar{\sigma}_n \\ \Phi \boldsymbol{\varepsilon}(n) + \Psi(\nu d(k) - n), & \sigma_n \wedge \bar{\sigma}_{n+1} \end{cases} \quad (3.12)$$

, ou Φ et Ψ sont les représentations discrétisées de F et f données par (3.7), avec le temps d'échantillonnage τ_s .

Le mode auxiliaire 3 est un mode de transition où S_1 s'ouvre avant la fin d'une sous période, notons que si l'on est dans ce mode alors la condition $\sigma_n \wedge \bar{\sigma}_{n+1}$ est vérifiée, impliquant que $\nu d(k) - n$ est compris entre zéro et un. D'où le troisième mode constitue une moyenne des deux autres.

L'erreur introduite par cette moyenne peut être minimisée en prenant ν grand.

On ajoute une contrainte à l'évolution des états $\boldsymbol{\varepsilon}(n)$, en imposant une limite de sécurité au courant d'induction.

$$-i'_{l,\max} \leq [\mathbf{1} \quad \mathbf{0}] \boldsymbol{\varepsilon}(n) \leq i'_{l,\max}, \quad n = 0, 1, \dots, \nu - 1. \quad (3.13)$$

La notion de ν -step model ne permet pas uniquement une bornitude du courant aux instants kT_s mais aussi aux instants d'échantillonnage intermédiaires avec une résolution de $\frac{T_s}{\nu}$.

En se basant sur la tension de sortie échantillonnée :

$$v_o(n) = g^T \varepsilon(n) \quad (3.14)$$

,on estime l'intégrale de l'erreur sur la tension de sortie par :

$$v'_{o,err} = \sum_{n=0}^{\nu-2} \frac{v'_o(n) + v'_o(n+1)}{2(\nu-1)} - v'_{o,ref} \quad (3.15)$$

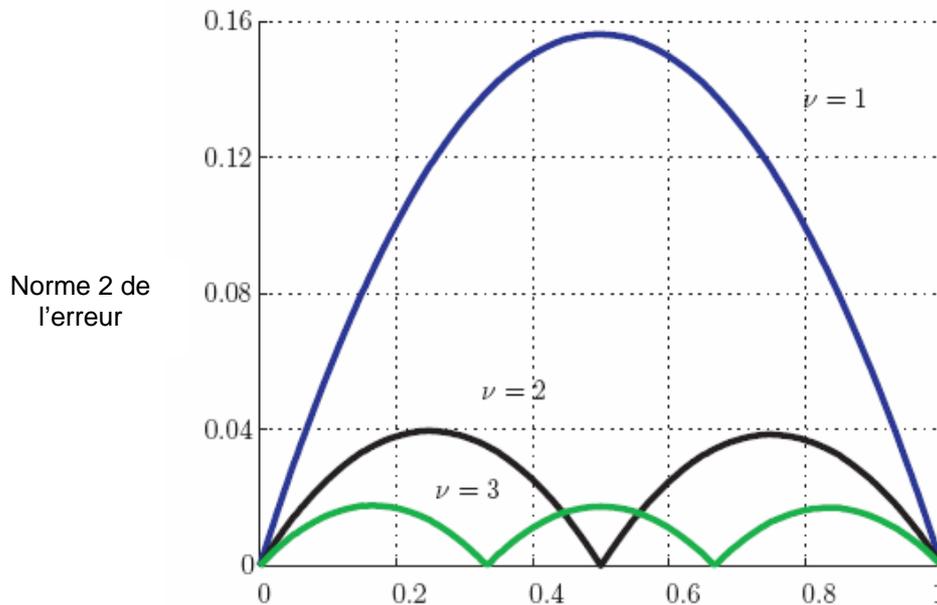


Figure 3.10 : Précision du ν -step model

En conclusion, l'approche du ν -step model permet une description de l'évolution des variables d'état le long d'une période. Tout particulièrement la variable discrète $\varepsilon(n)$, qui est une représentation échantillonnée précise de l'évolution continue de $x'(t)$ pour $t \in [kT_s, (k+1)T_s]$. La seule approximation qui a été faite apparaît dans l'introduction du mode auxiliaire lors de l'ouverture de S_1 .

Le paramètre ν est un paramètre de conception qui est choisi suivant la précision recherchée. Pour l'ensemble des paramètres du convertisseurs donnés dans le tableau 1, et qui seront utilisés lors des simulations on a tracé la norme 2 de l'erreur, pour plusieurs valeurs de ν suivant le rapport cyclique $d(k)$. Etant donné que l'erreur est indépendante du vecteur d'état, alors ces résultats sont valables sur tout l'espace d'état. Le choix de $\nu=1$ donne le modèle de moyenne standard, et qui était le plus utilisé pour la modélisation des convertisseurs DC-DC. Comme on peut le voir sur la figure ce modèle est parfait pour $d(k)=0$ et $d(k)=1$, et très mauvais pour $d(k)=0.5$, alors qu'on observe une nette amélioration pour seulement $\nu=2$, ou l'erreur diminue consécutivement et s'annule deux fois plus. Pour un modèle encore plus précis valable pour n'importe quel point de fonctionnement, on a pris $\nu=3$.

Avant de commencer la programmation ,on définit des contraintes sur les états ,les paramètres et l'entrée .Pour les états il faut que $x'(k) \in X' = [-i'_l, i'_l] \times [-v'_o, v'_o]$,et pour les paramètres , $[v'_{o,ref}, i'_{l,max}] \in V' = [v'_{o,ref}, v'_{o,ref}] \times [i'_{l,max}, i'_{l,max}]$,ou ses bornes dépendent de l'application ,quand à l'entrée elle est physiquement contrainte à être comprise entre zéro et un.

a. Le v -step modèle sous le formalisme MLD :

Les trois modes de fonctionnement du v -step modèle, permettent une modélisation adéquate du système en se basant sur l'aspect intrinsèque hybride de notre convertisseur. Afin de modéliser ces modes ,on a utilisé un nouveau langage évolué spécifique au systèmes hybrides ,le HYSDEL (Langage de Description des Systèmes Hybrides) ,le compilateur du HYSDEL tire automatiquement du programme les matrices correspondantes aux formalisme MLD donnée en (8.2) .Pour notre v -step model on obtient un modèle MLD à deux variables d'états ,deux paramètres ,(7n) z-variables ,(2(n-1)+1) δ -variables et (24n+8) contraintes sous forme d'inégalités mixed-integr linéaires.

b. Le v -step sous le formalisme PWA :

Afin d'implémenter la loi de commande du retour d'état explicite dans la MPT toolbox, le v -step model doit être sous forme d'un système hybride affine par morceaux ou (PWA) Piece Wise Affine, Un algorithme spécifique à la MPT toolbox permet une transformation efficace du MLD vers le PWA .Plus de détails sur le passage du formalisme MLD au formalisme PWA .

Le modèle PWA est défini comme un ensemble de dynamiques affines réparties dans un polyèdre d'un espace de dimension cinq $\Gamma = X' \times V' \times U$ généré par 2 états ,2 paramètres et une entrée. Alors que les variables binaires σ_n , $n = 0, \dots, v - 1$ définissent dans (3.11) les différents modes le long du rapport cyclique .On suppose que l'espace Γ est partitionné en v polyèdres suivant l'axe du rapport cyclique.

Exemple1 :

Afin de visualiser le PWA model du convertisseur ,considérons l'ensembles des paramètres du tableau 1 .De plus ,on pose $\Gamma = X' = [-4,4] \times [-0.1,1]$ p.u. , $V' = [0.2,1] \times [0.6,3]$ p.u., $U = [0,1]$. Pour observer la partition en 3 polyèdres, on rajoute une intersection avec les axes : $v'_{o,ref} = 0.556$ p.u. et $i'_{l,max} = 1.667$ p.u. ,ce qui élimine les deux dimensions correspondant à l'espace des paramètres V' .La figure 8.4 permet de voir la partition polyédrale de l'espace $X' \times U$,ou on a rajouté la contrainte $i'l' \in [-i'_{l,max}, i'_{l,max}]$. On peut noter qu'une faible tension au borne de la capacité et un rapport cyclique important ,induisent un courant d'induction élevé ,et vice versa .

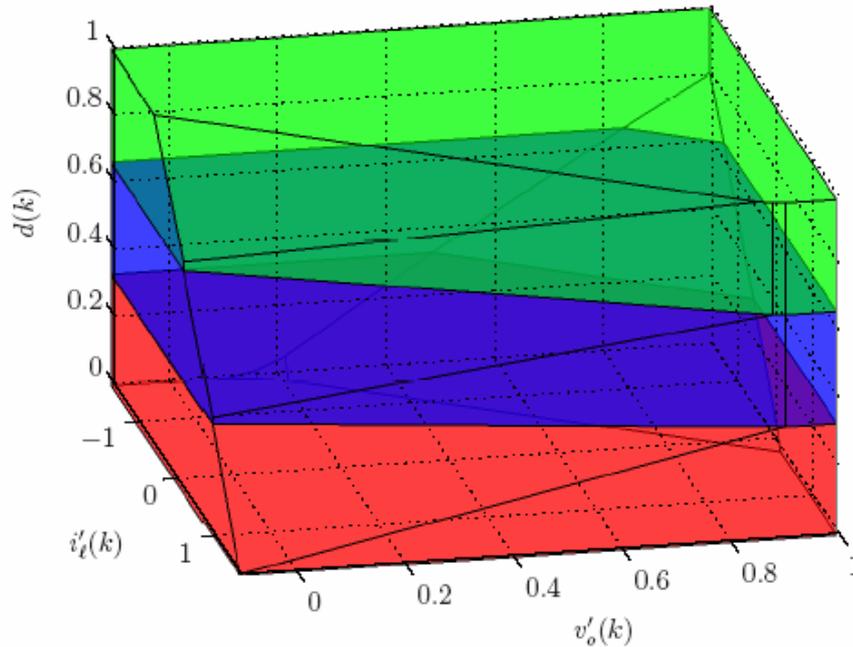


Figure 3.11 : Partition en polyèdres du modèle PWA du convertisseur pour les paramètres du tableau 1, et $v'_{o,ref} = 0.556 p.u.$ et $i'_{l,max} = 1.667 p.u.$

La figure 3.11 permet de voir clairement l'aspect hybride du convertisseur, on peut remarquer que les dynamiques sont linéaires par rapport aux variables d'état, car il n'y a pas de division au niveau de l'espace d'état. Alors qu'elles sont non linéaires par rapport au rapport cyclique. On a approximé cette non linéarité en introduisant le ν -step model, ou le rapport cyclique est divisé en trois segments, tel que la transition du mode 1 vers le mode 2 est modélisée par un troisième mode auxiliaire comme il a été dit plus haut. On constate aussi que ce modèle hybride est continu lors du passage d'un mode au mode voisin. Ce qui se déduit de l'équations d'état (3.12) et confirmée par la figure (3.3).

3.2. Commande optimale sous contrainte :

Dans cette partie, on définit les objectives de commande sous forme d'une fonction ou d'un critère qui associé au modèle MLD du convertisseur implémenté sous HYSDEL permettent de poser un problème de commande optimale avec contrainte en temps fini (CFTOC) exposé dans le chapitre précédent, implémenté sous la multi parametrics toolbox (MPT) de Matlab, qui peut être exécuté en temps réel ou en off line afin de tirer dans ce dernier cas la loi de commande qui lui est associée.

Par la suite, on suppose que :

- ✦ La tension d'entrée et de sortie v_s et v_o respectivement, et le courant d'induction i_l sont mesurables.
- ✦ La tension de référence $v_{o,ref}$ et le courant limite $i_{l,max}$ sont donnés par le cahier de charge.

Ce qui permet de tirer directement les entrées de notre problème v'_o , $v'_{o,ref}$, i'_l et $i'_{l,max}$.

3.2.1. Le critère à optimiser :

D'une manière générale, l'objectif de commande pour notre système, est la régulation de la valeur moyenne de la tension de sortie, vers sa référence, chose qui revient à :

- ✦ la minimisation de l'erreur sur la tension de sortie par rapport à sa référence $v_{o,err}(k)$, en conciliant vitesse et dépassement (i),
- ✦ malgré les variations de la tension d'entrée v_s ou celles de la résistance de charge r_0 (ii),
- ✦ tout en respectant les contraintes sur le courant d'induction (iii).

Pour l'instant on suppose que la résistance de sortie est connue.

Afin d'appliquer une commande stable sous un rapport cyclique constant non nul, on introduit :

$$\Delta d(k) = d(k) - d(k - 1) \quad (3.16)$$

, la différence entre deux rapports cycliques consécutifs.

Soit la matrice de contrainte sur la sortie $Q = \text{diag}(q_1, q_2)$ avec $q_1, q_2 \in \mathbf{R}^+$, et le vecteur de sortie $y(k) = [v'_{o,err}(k) \quad \Delta d(k)]^T$, avec $v'_{o,err}$ définie par l'expression (8.15), considérons le critère d'optimisation suivant :

$$J(x'(k), d(k - 1), v'_{o,ref}(k), i'_{l,max}, D(k)) = \sum_{l=0}^{N-1} \|Qy(k + l|k)\|_1, \quad (3.17)$$

qui contraint l'évolution prédictive $y(k + l|k)$ à l'instant k , le long du domaine de fini N en utilisant la norme 1. Il est clair que le critère ne dépend pas seulement de la séquence des entrées de commande $D(k) = [d(k), \dots, d(k + N - 1)]^T$ et de l'état courant (mesuré) $x'(k)$, mais aussi de la dernière entrée de commande $d(k-1)$, de la tension de référence $v'_{o,ref}$ et du courant limite $i'_{l,max}$, qui peuvent être variants dans le temps, estimant ainsi les variations de v_s .

Un bilan des objectifs fixés permet de voir que le premier objectif (i) est compris dans le modèle de prédiction, et le second (ii) est atteint grâce à la normalisation du modèle de prédiction par v_s , ce qui amène à alimenter le circuit par $v'_{o,ref}$, qui représente l'inverse de la tension v_s , quand à la résistance de sortie, elle est considérée constante, alors que le troisième objectif (iii) est exprimé comme une contraintes physique dans le modèle de prédiction sous sa forme MLD.

3.2.2. La détermination en temps réel de l'entrée de commande :

L'entrée de commande à l'instant k est obtenue en minimisant le critère (17), le long de la séquence de commande $D(k)$, sous les contraintes linéaires d'inégalités mixed-integer du modèle MLD décrit par (8.2), les contraintes matérielles (physiques) imposées aux séquences du rapport cyclique :

$$0 \leq d(l) \leq 1, l = k, \dots, k + N - 1 \quad (3.18)$$

et l'expression (3.16). Ce qui donne le CFTOC :

$$D^*(k) = \arg \min_{D(k)} J(x'(k), d(k-1), v'_{o,ref}(k), i'_{l,max}(k), D(k)) \quad (3.19.a)$$

$$\text{sous le modèle MLD, (3.18), (3.16)} \quad (3.19.b)$$

, induisant une séquence de rapport cyclique optimale $D^*(k)$, de laquelle uniquement le premier élément $d^*(k)$ est appliqué au convertisseur. Au prochain intervalle d'échantillonnage, k passe à $k+1$, un nouveau vecteur de mesure est récupéré, et le CFTOC est résolu une nouvelle fois suivant les règles de la technique du receding horizon présentée dans le chapitre précédent. Etant donné qu'on a utilisé la norme 1, pour tout les cas, le problème revient à résoudre un MILP (Mixed Integer Linear Program) pour lequel des solutions efficaces ont été trouvées. [ILO02].

3.2.3 Détermination off-line de la loi de commande :

Afin de permettre une implémentation du régulateur proposé, malgré la grande fréquence de commutation, la solution du CFTOC doit être calculée en off line. Pour cela, on remplace le modèle MLD par son équivalent PWA. Pour le modèle PWA, on suppose que la résistance de sortie r_0 est nominale, invariante dans le temps d'une valeur de 1p.u. On détermine grâce aux méthodes de programmations multiparamétrique et dynamique offerts par la MPT, la loi de commande du retour d'état de notre convertisseur, ou le vecteur d'état $x'(t)$, la dernière entrée de commande $d(k-1)$, la tension de sortie de référence $v'_{o,ref}$ et le courant limite $i'_{l,max}$, sont traités comme des paramètres.

Les paramètres sont rassemblés dans le vecteur $p'(k) = \left[(x'(k))^T \ d(k-1) \ v'_{o,ref}(k) \ i'_{l,max}(k) \right]^T$, avec $p'(k) \in \Pi = X' \times U \times V'$. La loi de commande résultante est une fonction affine par morceaux ou PWA de $p'(k)$, définie par une partition polyédrale de dimension cinq de l'espace Π .

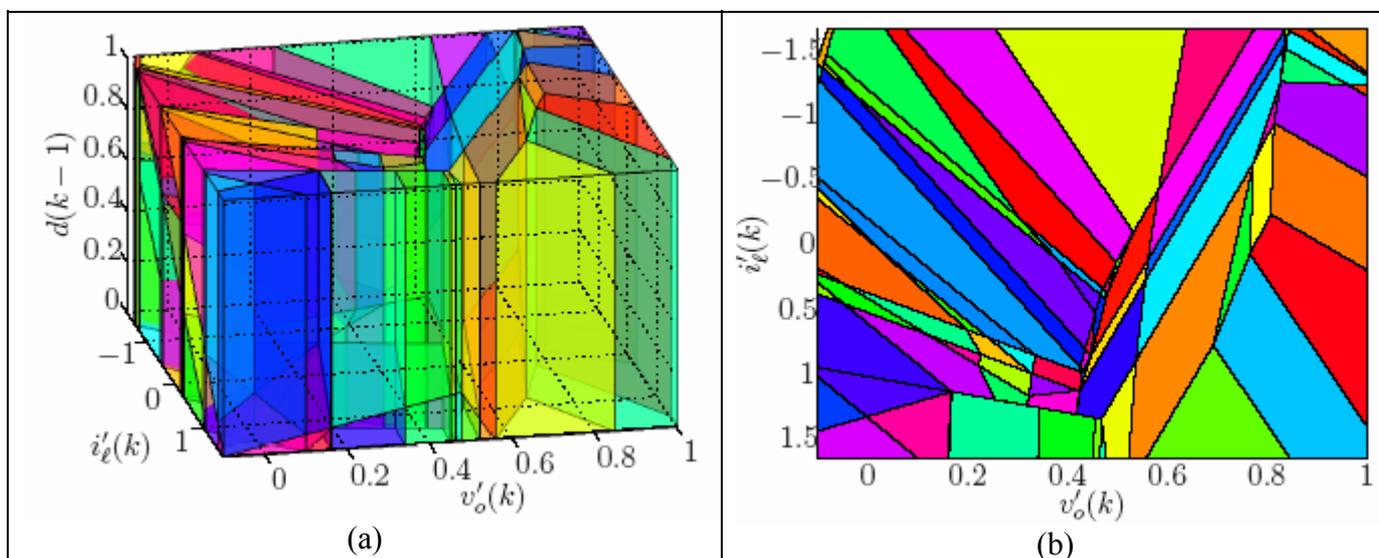


Figure 3.12 : répartition en polyèdres de loi de commande dans :
 a) l'espace $X' \times U$
 b) l'espace X'

Exemple 2 :

Pour le modèle PWA de l'exemple 8.1, et en utilisant les paramètres du tableau 1, on détermine la loi de commande du retour d'état qui est PWA, défini sur 633 polyèdres dans l'espace de dimension cinq Π . En utilisant des algorithmes d'optimisation de complexité, on obtient un régulateur simplifié à 121 régions ou polyèdres seulement.

Afin de visualiser la loi de commande du retour d'état, on pose $v'_{o,ref} = 0.556 p.u.$ et $i'_{l,max} = 1.667 p.u.$, ce qui nous donne une loi de commande dans les conditions de fonctionnement nominal, définie sur l'espace à trois dimensions $X' \times U$. Comme on peut le voir à la figure 8.5 qui représente la division en polyèdre, et à la figure 8.6 qui représente l'entrée de commande $d(k)$ en fonction du vecteur d'état $x'(k)$, ou on a ajouté la condition $d(k-1)=0.6$, ce qui nous ramène à un espace de deux dimensions.

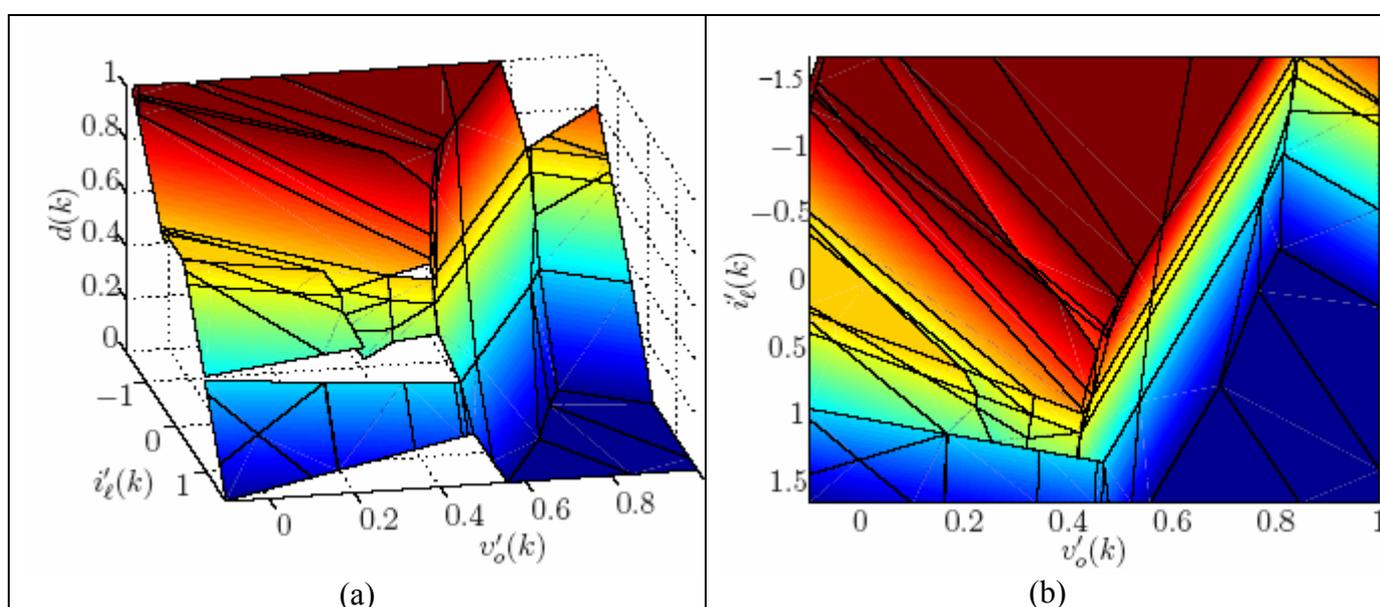


Figure 3.13 : loi de commande du retour d'état dans :
 a) l'espace $X' \times U$
 b) l'espace X'

3.3. Résultats de simulation :

Cette partie permet de voir l'efficacité de la méthode de commande proposée dans ce chapitre. Notre intérêt s'est surtout porté sur le comportement et l'étude de l'évolution du système lors du démarrage qui est un point important qui a posé bien des problèmes lors d'une utilisation pratique du convertisseur, les simulations sont basées sur une considération du modèle non linéaire du convertisseur comme modèle réel du système, quand à la boucle fermée elle est faite à base du retour d'état prenant en compte le modèle PWA du convertisseur. Le courant d'induction l'entrée et les tensions d'entrée et de sortie sont considérées mesurables directement. Les variables représentées dans les figures suivantes sont toutes exprimées en per unit et une unité de l'axe des temps correspond à une période de commutation.

Les paramètres du modèle non linéaire ont été choisis de telle sorte qu'il représente un convertisseur abaisseur réel de 24 V à 12 V, avec une puissance de 150 W, pour cela on prend les valeurs données dans le tableau 1, pour une tension d'entrée de $v_s = 1.8$ V, une résistance de charge $r_o = 1$ p.u. et une tension de sortie de référence $v_{o,ref} = 1$ p.u. Quand au modèle de prédiction il utilise les mêmes paramètres que ceux du modèle non linéaire, sauf qu'ils sont normalisés par rapport à la tension d'entrée v_s .

PARAMETRES DU CONVERTISSEURS		
$x_c = 70$ p.u	$x_l = 3$ p.u	$i_{l,max} = 3$ p.u
PARAMETRES DU REGULATEUR		
N= 3 $v=3$	$q_1=4$	$q_2=4$

Tableau 1. Les paramètres de Simulation

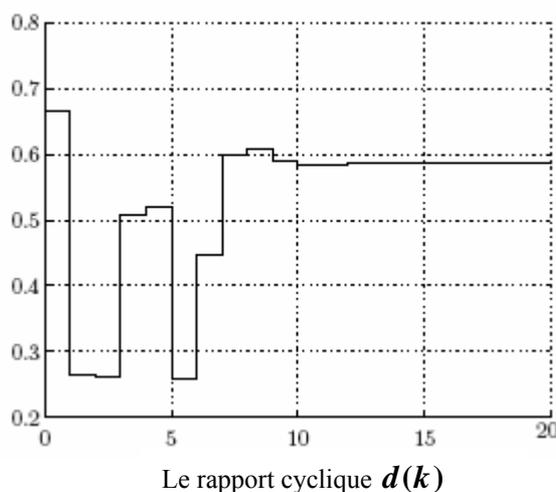
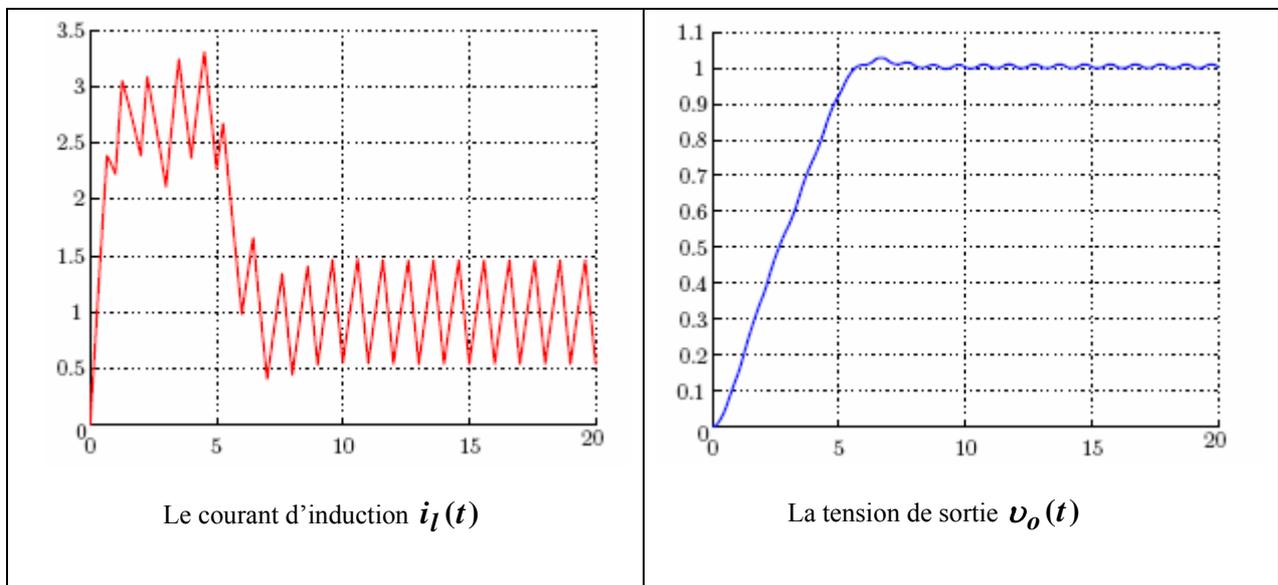


Figure 3.14 : simulation de la boucle fermée dans les conditions de travail nominales pendant le démarrage.

Cette figure représente la réponse indicielle du convertisseur aux conditions de travaux nominales avec l'état initiale $x(\mathbf{0}) = [\mathbf{0} \ \mathbf{0}]^T$ et $d(-1) = \mathbf{0}$, la tension de sortie se stabilise après 10 périodes de commutations, avec un dépassement inférieur à cinq pour cent, on peut voir aussi que la contrainte sur le courant est généralement respectée, l'erreur sur ce dernier dépendant totalement du choix du nombre de sous périodes imposé par le ν -step model peut être réduit en augmentant ce dernier.

Conclusion :

On a présenté dans ce chapitre une nouvelle méthode de modélisation et de commande d'un type de systèmes hybrides en prenant comme exemple le buck DC-DC, ou l'approche révolutionnaire du ν -step model, a permis une modélisation adéquate du système prenant en compte toutes les dynamiques et les caractéristiques intrinsèques du convertisseur contrairement à la modélisation moyenne utilisée précédemment, quand à la commande, un problème de commande optimale avec contraintes en temps fini a été formulé on line et off line afin de réaliser les objectifs de commande, en prenant en compte les contraintes physiques sur le système. Un retour d'état sous forme d'une partition de polyèdre a été obtenu grâce à la commande optimale off line, ce dernier a permis une implémentation simple sous la multi paramétrique toolbox de Matlab, ou ont été pris comme paramètres non seulement les variables d'état $i'_l(k)$ et $v'_o(k)$ et le rapport cyclique $d(k)$ mis à l'échelle par rapport à la tension d'entrée, mais aussi la tension de sortie de référence $v'_{o,ref}(k)$ et le courant limite $i'_{l,max}(k)$, cette mise à l'échelle permet de rejeter facilement les perturbations sur la tension d'entrée. Comme on a pu le voir sur les résultats de simulation notre régulateur permet une dynamique stable du système.

Conclusion Générale :

Ce travail permet une initiation à un domaine qui connaît un grand essor ces dernières années, ne cessant d'évoluer afin d'améliorer la compréhension et la maîtrise des systèmes et processus qui nous entoure et qui sont en général tous de nature hybride ,premièrement la partie généralité nous a permis de situer cette nouvelle notion de systèmes hybrides par rapport à nos acquis sur les systèmes classiques en donnant les différentes représentations possibles dont les plus récentes la représentation mixte linéaire entier et la représentation affine par morceaux qui permettent une prise en compte total des caractéristiques intrinsèques des systèmes hybrides ensuite le chapitre deux, ou est regroupée un étude sur la stabilité, l'observateur et la commande des S.D.H. Pour le premier point on a résumé les principaux travaux effectués sur la stabilité de ces systèmes ,et plus particulièrement la classe des systèmes à commutation de modèles qui a connu un intérêt tout particulier, puisqu'elle est la plus répandue et la plus simple à étudier ,cette étude nous a permis de constater qu'il n'est pas toujours simple de conclure à la stabilité pour ces systèmes, et comme on a pu le voir dans le point suivant cet intérêt pour les systèmes commutés ne se restreint pas à la stabilité, car des observateurs particuliers ont été conçu pour ce type de systèmes, dans notre cas on s'est basé sur une adaptation des observateurs à mode glissant aux systèmes hybrides, évidemment il était indispensable afin d'avoir une étude générale de toucher au point de la commande qui est celui qui suscite le plus d'intérêt et de difficultés, là aussi différentes classes de commandes ont été utilisées, mais on s'est restreint à celle qui nous a paru la plus efficace jusqu'à maintenant, la commande optimale, ou on a résumé les principaux travaux effectués sur ce sujet et exposé une approche particulière implémentée par la suite à laide de la programmation multiparamétrique, et dans le but d'éclaircir et de consolider ces nouvelles notions acquises tout au long de ce travail, on a conclue par deux applications, la première hydraulique, représente une régulation de niveau d'un banc d'essai de deux réservoirs, puis la deuxième sur le convertisseur DC-DC, qui est un exemple type de systèmes hybrides à commutation de modèle,ou on a d'abord modélisé le système dans le domaine continue ,puis on a introduit la notion du v -step model qui répond totalement aux objectifs posés, ensuite ce modèle est exprimé sous le formalisme de la représentation MLD pour l'étude de l'aspect hybride puis sous le formalisme de la représentation PWA pour la commande, l'étude a été faite en se basant sur les principes de la programmation multiparamétrique de la MPT toolbox de Matlab et du langage pour systèmes hybrides le Hysdel.

Perspectives :

Etant donné que ce travail représente le premier effectué sur les systèmes hybrides au sein de l'école, on espère le compléter dans des travaux futurs par :

1. Elargissement de la classe des systèmes hybrides considérés (incertains, avec retard, zénon...).
2. Etude d'autres techniques de commande que la commande optimale.
3. Application d'un observateur à modes glissants d'ordre supérieur (super twisting) afin d'éliminer le phénomène de broutement.
4. Pour l'application du convertisseur effectuer une étude de robustesse de la commande utilisée pour des variations de la tension d'entrée, et de la résistance de charge.

Annexe A

Dans cette partie, nous rappelons les principales caractéristiques [Cas93] des systèmes que nous allons étudier. Nous allons voir qu'à partir de ces caractéristiques et notamment à partir de la notion de variable d'état, on peut classer ces systèmes en différents types. Ainsi, avec cette classification, et d'après la définition 1.1, nous allons mieux identifier les systèmes qui nous intéressent dans ces travaux.

Caractéristiques des systèmes :

- **Systèmes dynamiques et systèmes statiques :** Les systèmes statiques ont une sortie qui est toujours indépendante des valeurs passées de son entrée. Les systèmes dynamiques ont une sortie qui dépend des valeurs passées de son entrée. Les équations différentielles ou aux différences sont requises pour décrire le comportement des systèmes dynamiques.
- **Systèmes stationnaires et systèmes non stationnaires :** Le comportement des systèmes stationnaires ne change pas avec le temps. Cela implique qu'à une entrée spécifique appliquée au système correspondra toujours la même sortie.
- **Systèmes linéaires et systèmes non linéaires :** Un système linéaire satisfait la condition suivante :
$$f(\lambda x + \mu y) = \lambda f(x) + \mu f(y),$$
Où x et y sont des vecteurs d'entrée, λ et μ des scalaires et $f(.)$ la sortie du système.
- **Systèmes avec des variables d'état continues et systèmes avec des variables d'état discrètes :** Les variables continues prennent leurs valeurs sur l'ensemble des réels \mathbb{R} , alors que les variables discrètes prennent leurs valeurs dans un ensemble fini de valeurs distinctes, par exemple un sous-ensemble des entiers naturels \mathbb{N} .

Le temps est une variable particulière qui peut également être soit continue, soit discrète. De plus, dans le temps discret, on distingue deux autres cas : les systèmes échantillonnés et les systèmes à événements discrets.

- **Systèmes à temps continu, systèmes échantillonnés, systèmes à événements discrets :** Les systèmes à temps continu changent d'état continûment lorsque le temps change. Les systèmes à temps discret ne changent d'état qu'à des instants précis. Lorsque ces instants sont synchrones, réglés par une horloge, on dit que ces systèmes sont échantillonnés. Lorsque ces instants sont asynchrones, on les appelle des événements. On définit alors les systèmes à événements discrets comme étant des systèmes à états discrets pilotés par des événements.

La figure ci-dessous représente une classification des systèmes d'après les caractéristiques décrites ci-dessus. Dans cette classification, on identifie cinq types de systèmes :

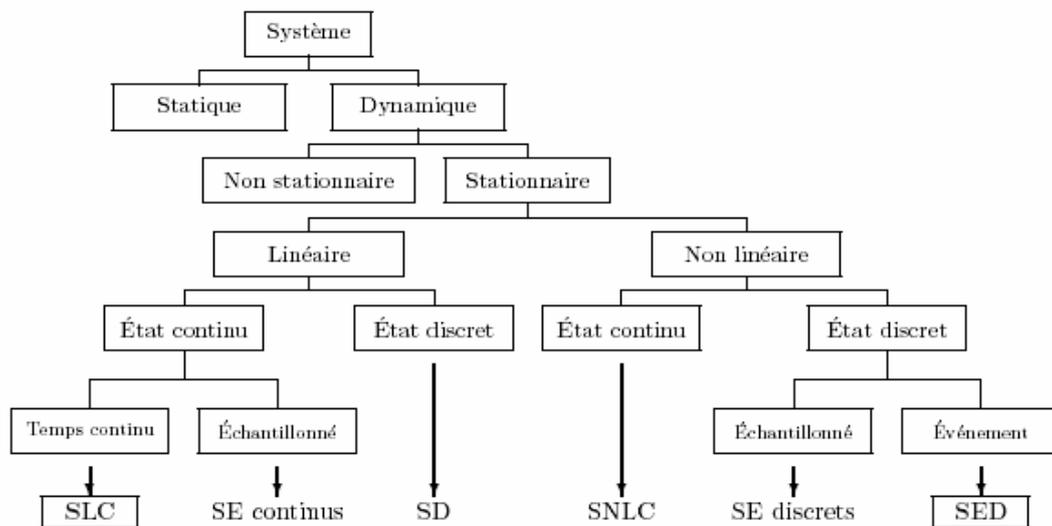
- Les systèmes linéaires continus notés SLC. Ces systèmes peuvent être modélisés par des équations différentielles.

- Les systèmes échantillonnés à états continus sont notés SE continus et ceux à états discrets sont notés SE discrets. Ces systèmes peuvent être modélisés par des équations aux différences.

- Les systèmes discrets notés SD. L'évolution des variables d'état peut se faire par exemple à partir de fonctions booléennes paramétrées dans le temps.

- Les systèmes non linéaires continus notés SNLC.

- Les systèmes à événements discrets notés SED. Ces systèmes peuvent être modélisés par des automates, des réseaux de Pétri ou l'algèbre (max,+).



Annexe B

Une représentation implicite pour un polyèdre \mathbf{p} donné par des contraintes s'exprime de la façon suivante :

$$p = \left\{ x_p \mid A_p x_p = b_p, C_p x_p \geq d_p \right\}$$

où A_p, C_p , sont des matrices et b_p, d_p et x_p , sont des vecteurs.

Le polyèdre \mathbf{p} admet alors une représentation paramétrique équivalente duale :

$$p = \left\{ x_p \mid x_p = L_p \lambda_p + R_p \mu_p + V_p \nu_p \right\}$$

exprimée comme une combinaison linéaire d'arêtes (matrice L_p), une combinaison convexe de sommets (matrice V_p), et une combinaison de rayons extrêmes (matrice R_p).

Annexe C

Généralités :

Stateflow est destiné à la représentation des systèmes à événements discrets et à la simulation de leur fonctionnement .Il communique parfaitement avec Simulink dont il renforce la capacité de prise en compte des séquences logiques .D'ailleurs, le bloc Stateflow n'est autre qu'un bloc Simulink masqué. Stateflow s'utilise de manière analogue à Simulink .On construit tout d'abord le schéma en plaçant, grace à un éditeur, les différents éléments du diagramme (chart) à construire et qui sont essentiellement :

- Les états (states),
- leurs liaisons (transitions)
- ainsi que les libellés attribués aux événements (events) et aux actions.

Comme pour les schémas Simulink, cette représentation graphique recouvre en fait la construction d'une S-fonction. Le schéma Stateflow étant inclut dans un modèle Simulink, il faut lier donc les deux programmes sous-jacents. Cette opération est réalisée automatiquement par la commande start du menu simulation. Cette commande lance la construction préalable du programme globale (langage C) associé au modèle ,sa compilation et enfin son exécution . Ce programme peut être téléchargé sur une cible temps réel et ainsi faire fonctionner un automate au toolbox Real Time Workshop.

La terminologie utilisée :

- Une conception hiérarchisée :

Un schéma stateflow doit être conçu de manière hiérarchisée .Les objets graphiques et non graphiques vont s'emboîter selon la hiérarchie parent-enfant .

1. Le plus haut niveau de la hiérarchie Stateflow est appelé « Machine », il correspond au niveau « Modèle » dans Simulink. Ce vocable est issu de concept Machine à états finis qui à été développé pour rendre compte du fonctionnement des systèmes à événements discrets. La machine peut réunir plusieurs diagrammes Stateflow dont elle est le parent.
2. Le diagramme (chart) : c'est le deuxième niveau qui correspond au sous système Simulink , décrit une partie de la machine. Il rassemble des états reliés par des transitions .Le diagramme décrit le fonctionnement logique d'un sous ensemble particulier de la machine.
3. L'état (state) : ce niveau correspond à l'état qu'on décrit par un vecteur d'état dans Simulink .Il représente l'état dans lequel se trouve la machine, en quelque sorte l'état actuel de son fonctionnement, par exemple l'état On dans l'exemple du thermostat .Un état peut contenir des sous-états , il devient alors super-état . Par exemple l'état marche d'une machine peut contenir deux sous-états, avant et arrière .L'état avant peut lui aussi avoir des enfants : rapide, lent, attente .Il n'y a pas de restrictions au nombre total de générations dans la descendance d'un parent.
4. L'événement (event) : cet objet stateflow est un objet non graphique (il n'apparaît sur le diagramme que grâce à son libellé). L'événement joue un rôle majeur dans Stateflow. C'est lui qui active un diagramme qui «s'est endormi »en l'absence d'événements. C'est lui qui va déclencher le franchissement d'une transition particulière d'un état vers l'état suivant et assurer l'exécution des actions.

- Des objets spécifiques :

Les objets Stateflow se rangent en deux catégories : des objets graphiques, représentés par un dessin symbolique normalisé et des objets non graphiques accessibles par leur label. Les informations qui concernent tous ces objets sont recueillies dans un dictionnaire appelé data dictionary .Les objets dont nous avons besoin pour construire un diagramme sont :

- l'état, représenté par un rectangle aux coins arrondis ;
- la transition, représenté par un lien orienté entre deux états .La transition défaut est particulière, elle indique l'état qui devra être actif à l'initialisation du diagramme ;
- des objets non graphiques qui sont les libellés donnés aux états, aux transitions, aux actions, aux événements....,ces noms sont associés à des variables manipulées par le programme (S-function) sous-jacent et ne sont donc pas de simples étiquettes de reconnaissance .

- Les données :

Les datas sont des objets non graphiques intervenant dans un diagramme en tant que données (obligatoirement numériques). A la différence des événements dont le rôle est de déclencher l'évolution du diagramme, les datas peuvent être :

- le résultat de cette évolution quand ils sont fabriqués par des actions ;
- la cause de cette évolution quand ils interviennent dans les conditions.

Un bloc Stateflow communique bien avec les blocs Simulink par l'intermédiaire des objets datas. Il reçoit de Simulink des données et il fournit à Simulink un résultat.

Tout comme pour un événement, une donnée doit être déclarée dans le dictionnaire. Si data est déclarée local elle ne sera visible que dans le diagramme ou l'état ou le sous-état dans le quel elle a été déclarée.

En pratique, on communique avec Simulink en déclarant des datas Input from ou Output to Simulink .Dans ce cas le bloc Stateflow est automatiquement pourvu d'un port (entrant ou sortant)qui sera relié au bloc Simulink approprié .

Si data est déclarée Constant, elle prend et garde la valeur indiquée dans Initial Value. Si elle est déclarée Workspace on doit l'affecter depuis Command window avant de lancer la simulation.

Annexe D

LANGAGE HYSDEL

But – Principe

La transformation de la description initiale d'un système hybride en une forme MLD exige l'application d'un ensemble de règles données par le calcul propositionnel. Il est maintenant nécessaire d'adopter un outil qui permette d'obtenir cette forme MLD de façon systématique, évitant ainsi des calculs fastidieux. Pour cela, un compilateur est développé, fournissant après calcul les matrices $E_1, E_2, E_3, E_4, E_5, D_1, D_2, D_3, C, B_1, B_2, B_3$ et A des équations (1.9)-(1.11). Le langage spécifiant le problème au compilateur se nomme HYSDEL (Hybrid System DEscription Language) (Torrise *et al.*, 2000). L'appel au compilateur se fait sous Matlab, par l'intermédiaire d'un fichier source « *.hys ». Il résulte de la compilation un fichier « *.m » dans lequel se trouve la forme MLD complète du système. Ce fichier regroupe donc sous la formalisation d'un « Objet Matlab » toutes les matrices de la forme MLD. Le fichier final inclut également un certain nombre de variables caractérisant les dimensions des matrices.

Présentation du langage HYSDEL

HYSDEL est un langage permettant de traiter un système hybride selon la structure MLD à partir d'une description textuelle de ce système. Chaque fichier est en réalité composé de deux parties. La première partie, appelée « INTERFACE » a pour but de déclarer toutes les variables : les états, les entrées, les sorties, ainsi que toutes les constantes. La deuxième partie, dite « MISE EN OEUVRE », est composée de six sections spécialisées décrivant les relations existant entre les variables, de la façon suivante :

1. **Section « AD »** : elle définit les variables auxiliaires logiques issues de l'interprétation de relations continues.

On trouvera ainsi dans cette section des relations : $\delta = 1$ si $a_1 x_1 + a_2 x_2 \leq b$.

2. **Section « DA »** : elle définit les variables auxiliaires continues, issues de la valeur prise par des variables booléennes.

On trouvera alors des relations : si $\delta = 1$ alors $z = a_1 x + b_1$ sinon, $z = a_2 x + b_2$

3. **Section « LOGIC »** : elle définit les fonctions impliquant des variables booléennes.

Ainsi on pourra y inclure des relations du type : $\{\delta = \delta_1 \& (\delta_2 | \delta_3)\}$

4. **Section « CONTINUOUS »** : elle décrit la dynamique linéaire, sous la forme d'équations aux différences.

Notons par exemple : $h(k) = h(k-1) + \frac{T_s}{A}(Q_1 - Q_2)$

5. **Section « AUTOMATA »** : cette section spécifie les équations caractérisant les transitions des états de l'automate.

Ainsi on y trouvera des relations : $V = (x_1 \& a) | (x_2 \& b)$.

6. **Section « MUST »** : cette section spécifie les contraintes sur les variables continues et binaires (contraintes linéaires et formules booléennes).

Par exemple : $x_{\min} \leq x \leq x_{\max}, \{(u_1 \& u_2) | (\bar{u}_1 \& u_2)\}$.

Enfin le modèle MLD obtenu n'est pas unique, dépendant de la façon d'introduire les variables auxiliaires. Il existe en effet de multiples formes MLD pour un même système, il est alors préférable de rechercher celui qui procure le nombre de variables logiques le plus faible. Malgré tout, cet outil fournissant automatiquement des modèles MLD s'avère très utile pour la définition de la forme d'état associée.

Annexe E

La commande optimale des systèmes linéaires par morceaux (PWA) ou avec contraintes suscite par sa facilité de formulation et de résolution de problèmes complexes, un grand intérêt auprès de la communauté de chercheurs en automatisme. Le rôle de la Multi Parametric Toolbox (MPT) est de fournir des moyens d'estimation efficaces, permettant d'obtenir sous Matlab des régulateurs en boucle fermée appropriés à ce type de problèmes de commandes optimales avec contraintes. Un problème linéaire ou quadratique d'optimisation est résolu off line grâce à la programmation multi paramétrique, dont la solution est sous forme d'une loi de commande d'un retour d'état PWA, c'est-à-dire que l'espace d'état est partitionner en ensemble de polyèdres, ou pour chaque ensemble la loi de commande optimale sous forme d'une fonction piece wise affine des états. L'une des raisons qui ont fait la popularité de cette boite à outil est que dans l'implémentation on line de tels régulateurs, l'estimation des valeurs du régulateur se réduit à un test d'adhésion à un ensemble prédéfini.

La Multi-Parametric Toolbox (MPT) est une nouvelle boite à outil de matlab pour la modélisation, l'analyse et la réalisation de régulateurs optimales pour les systèmes linéaires, non linéaires et hybrides avec contraintes, afin de garantir l'efficacité de cette boite à outil, une bibliothèque étendue d'algorithmes sur la géométrie d'estimation et l'optimisation multi paramétrique sont disponibles.

La boite à outils offre un large spectre d'algorithmes compilés dans un format simple et facilement accessible : démarrant par la réalisation de différents objectifs de performances (linéaire, quadratique et temps minimum) jusqu'à la manipulation de systèmes avec des incertitudes sous forme de polytopes.

1. La Modélisation des systèmes dans la MPT :

Il est possible de modéliser directement les dynamiques de deux types de systèmes :

1. Les systèmes linéaires invariants dans le temps (LTI) :

Ces systèmes sont de la forme :

$$\begin{aligned} \mathbf{x}(k + 1) &= \mathbf{A}\mathbf{x}(k) + \mathbf{B}\mathbf{u}(k) \\ \mathbf{y}(k) &= \mathbf{C}\mathbf{x}(k) + \mathbf{D}\mathbf{u}(k) \end{aligned} \quad (1)$$

, ils sont stockés en utilisant :

```
sysStruct.A = A  
sysStruct.B = B  
sysStruct.C = C  
sysStruct.D = D
```

2. Les systèmes linéaires par morceaux (Piece Wise Affine) :

Ces systèmes sont de la forme :

$$\mathbf{x}(k + 1) = \mathbf{A}_i \mathbf{x}(k) + \mathbf{B}_i \mathbf{u}(k) + \mathbf{f}_i \quad (2)$$

$$\mathbf{y}(k) = \mathbf{C}_i \mathbf{x}(k) + \mathbf{D}_i \mathbf{u}(k) + \mathbf{g}_i \quad (3)$$

$$si \begin{bmatrix} \mathbf{x}(k) \\ \mathbf{u}(k) \end{bmatrix} \in D_i \quad (4)$$

Où l'équation (4) définit une partition D en polyèdres de l'espace des états et des entrées, délimitant les zones ou les différentes dynamiques sont actives. Les segments de cette partition sont définis en utilisant les guard lines c'est-à-dire des contraintes sur les entrées et les états, définis sous la forme suivante :

$$\mathbf{G}_i^x \mathbf{x}(k) + \mathbf{G}_i^u \mathbf{u}(k) \leq \mathbf{G}_i^c \quad (5)$$

La dynamique i est active si la contrainte (5) est réalisée. Ce système est défini sous la MPT en utilisant :

```

sysStruct.A = {A1, A2, ..., An}
sysStruct.B = {B1, B2, ..., Bn}
sysStruct.f = {f1, f2, ..., fn}
sysStruct.C = {C1, C2, ..., Cn}
sysStruct.D = {D1, D2, ..., Dn}
sysStruct.g = {g1, g2, ..., gn}
sysStruct.A = {A1, A2, ..., An}
sysStruct.guardX = {Gx1, Gx2, ..., Gxn}
sysStruct.guardU = {Gu1, Gu2, ..., Gun}
sysStruct.guardC = {Gc1, Gc2, ..., Gcn}

```

Comme il est possible d'importer des modèles de différentes sources, la MPT permet de modéliser les lois de commande des systèmes linéaires discrets, des systèmes à commutations et des systèmes hybrides. Les systèmes hybrides peuvent être décrits sous le formalisme PWA ou le formalisme MLD, en sachant qu'il existe des algorithmes efficaces permettant de passer d'une représentation à l'autre, les sources de modèles peuvent être :

- ✦ Une structure MLD générée par `mpt pwa2mld`.
- ✦ Des modèles non linéaires définis par `mpt nonlinfcn`.
- ✦ Des fonctions d'état et des fonctions de transfert.
- ✦ Des objets de la boîte à outil MPC.
- ✦ Des objets de la boîte à outil identification des systèmes.

Afin d'importer ces types de systèmes on utilise :

```
model = mpt_sys (objets , Ts )
```

L'un des langages les plus associés à la MPT est le `hysdel`, ou est utilisée la fonction suivante pour utiliser les systèmes définis sous ce langage :

```
sysStruct = mpt_sys ('hysdelfile.hys', Ts)
```

le système est directement converti sous le formalisme PWA, afin de le garder sous le formalisme MLD on utilise :

```
sysStruct = mpt_sys ('hysdelfile.hys', Ts, 'mld')
```

La MPT permet aussi de définir différents types de contraintes :

- ✦ Les contraintes sur le Min/Max des sorties.
- ✦ Les contraintes sur le Min/Max des variables à manipuler.
- ✦ Les contraintes sur le Min/Max des vitesses variables des variables à manipuler.
- ✦ Les contraintes sous forme de polyèdres sur les états.

D'où on définit une **sysStruct** qui est la forme sous laquelle est stocké un système afin d'être commandé pour un système PWA par :

```

sysStruct.A = {A1, ..., An}
sysStruct.B = {B1, ..., Bn}
sysStruct.C = {C1, ..., Cn}
sysStruct.D = {D1, ..., Dn}
sysStruct.f = {f1, ..., fn}
sysStruct.g = {g1, ..., gn}
sysStruct.guardX = {guardX1, ..., guardXn}
sysStruct.guardU = {guardU1, ..., guardUn}
sysStruct.guardC = {guardC1, ..., guardCn}
sysStruct.ymax = ymax;

```

```

sysStruct.ymin = ymax;
sysStruct.umax = umax;
sysStruct.umin = umin;
sysStruct.dumax = dumax;
sysStruct.dumin = dumin;

```

2.La commande dans la MPT:

Pour les systèmes linéaires et hybrides sous contraintes, la MPT permet de construire des lois de commande optimales ou sous optimales, soit on line ou est résolu un problème d'optimisation à chaque instant bouclé à l'aide de la technique de Receding Horizon Control (RHC), soit en utilisant off line la programmation multi paramétrique qui permet d'avoir une représentation explicite de la loi de commande, dont la syntaxe est :

```
ctrl = mpt_control (sysStruct, probStruct)
```

Quand à la commande on line, elle peut être générée en utilisant :

```
ctrl = mpt_control (sysStruct, probStruct, 'online')
```

ou sysStruct et probStruct sont définis en détails dans le manuel de la MPT. La fonction principale appelle automatiquement l'une des fonctions du tableau 1. avant de calculer la solution optimale d'un problème de commande la fonction mpt_control vérifie d'abord si toutes les données de sysStruct et probStruct ont été déclarées, sinon la procédure est arrêtée et un message d'erreur est affichée. Une fois la loi de commande calculée, elle est retournée sous forme d'un objet mptctrl.

Système	N	Sous optimalité	Problème	Fonction
LTI	Fixé	0	CFTOC	mpt_optControl
LTI	Inf	0	CITOC	mpt_optInfControl
LTI	Inf	1	CMTOC	mpt_iterative
LTI	Inf	2	Low Comp	mpt_oneStepCtrl
PWA	Fixé	0	CFTOC	mpt_optControlPWA
PWA	Inf	0	CITOC	mpt_optInfControlPWA
PWA	Inf	1	CMTOC	mpt_iterativePWA
PWA	Inf	2	Low Comp	mpt_iterativePWA

Tableau 1 : Liste des différentes stratégies appliquées pour différents problèmes

La MPT fournit une variété de sous programmes appelés à partir de la fonction mpt_control. L'un des problèmes de commande est alors résolu pour des conditions spécifiques sur le système et le problème d'optimisation :

1. Commande Optimale avec Contraintes en Temps Fini (CFTOC).
2. Commande Optimale avec Contraintes en Temps Infini (CITOC).
3. Commande Optimale en Temps Minimal avec Contraintes (CMTOC).
4. Complexité réduite.

Pour un critère à optimiser :

$$J_N = \sum_{k=0}^{N-1} \left(\|Q(x_k - x_{ref})\|_p + \|R(u_k - u_{ref})\| \right)$$

on définit les éléments d'une probStruct :

✦ L'horizon de prédiction :

$$\text{probStruct.N} = \{ N \mid \text{Inf} \}$$

✦ Type de la norme à utiliser :

$$\text{probStruct.norm} = \{ 1 \mid \text{Inf} \mid 2 \}$$

✦ Signaux de références :

$$\text{probStruct.} \{ \text{xref} \mid \text{uref} \mid \text{yref} \mid \text{dref} \mid \text{zref} \}$$

✦ Les matrices de contraintes :

$$\text{probStruct.} \{ Q \mid R \mid Qy \mid Qd \mid Qz \mid Rdu \}$$

pour plus de détails se référer au manuel.

3. L'analyse des systèmes dans la MPT :

La MPT possède plusieurs fonctions pour l'analyse des systèmes hybrides et l'étude de la réalisabilité et la stabilité des lois de commande obtenues par les différentes commandes. La fonction de Lyapunov d'un régulateur peut être obtenue par :

$$\text{ctrl} = \text{mpt_lyapunov}(\text{ctrl}, \text{type})$$

tel que type renvoi à :

- ✦ Quadratique
- ✦ Somme de carrés
- ✦ Piecewise Affine
- ✦ Piecewise Quadratic
- ✦ Piecewise Polynomial

Quand à l'espace atteignable il est obtenu par :

$$\text{mpt_reachSets}$$

D'autres fonctions plus détaillées sont disponibles dans le manuel.

4. Visualisation dans la MPT :

Il est possible de visualiser la partition en polyèdres du régulateur off line en utilisant :

$$\text{mpt_plot}(\text{ctrl})$$

, comme il est possible de visualiser uniquement la commande en utilisant :

$$\text{mpt_plotU}(\text{ctrl})$$

, ainsi que les valeurs du critère par :

$$\text{mpt_plotJ}(\text{ctrl})$$

La fonction :

$$\text{simplot}(\text{ctrl})$$

, permet de visualiser la boucle fermée pour la valeur initiales zéro, pour une valeurs initiales différentes on utilise :

$$\text{simplot}(\text{ctrl}, x_0)$$

5.L'interface de la MPT :

Une interface simple et efficace a été conçue afin de faciliter l'utilisation de la mpt sans passer par l'appel des différentes fonctions. Pour accéder à cette interface il suffit de taper après l'installation et l'initialisation :

mpt_studio

Elle permet de déclarer directement les paramètres d'un système ou de les importer d'une source prédéfinie, de définir les contraintes et les limites sur les différents paramètres ou variables, pour concevoir un régulateur on line ou off line avec toutes les possibilités décrites dans la partie commande, puis de visualiser la partition en polyèdres de la solution optimale, la boucle ouverte ou fermée, la commande, le critère, elle permet aussi de construire une fonction de lyapunov pour le régulateur et une réduction de la complexité de la solution obtenue.

6. La programmation multi paramétrique :

Considérons le problème d'optimisation basique :

$$\begin{aligned} \min_{\zeta} f_o(\zeta) \\ \text{sous } f_i(\zeta) \leq 0 \quad i = 1, \dots, n_f \end{aligned} \quad (6)$$

Ou la variable à optimiser $\zeta = \begin{bmatrix} \zeta_r \\ \zeta_b \end{bmatrix}$ contient une partie réelle $\zeta_r \in \mathbf{R}^{d_r}$, et une partie binaire $\zeta_b = \{0,1\}^{d_b}$ avec $d = d_r + d_b$, le problème revient à la détermination d'un ζ minimisant la fonction coût $f_o : \mathbf{R}^d \longrightarrow \mathbf{R}$ tel que les contraintes inégalités $f_i(\zeta) \leq 0, i = 1, \dots, n_f$ soient vérifiées ou la fonction $f_i : \mathbf{R}^d \longrightarrow \mathbf{R}$ est appelée fonction de contrainte inégalité .

Reconsidérons maintenant le problème (6) en ajoutant un paramètre apparaissant au niveau des fonctions coût et contraintes qui deviennent $f_o : \mathbf{R}^d \times \mathbf{R}^{n_r} \longrightarrow \mathbf{R}$ et $f_i : \mathbf{R}^d \times \mathbf{R}^{n_r} \longrightarrow \mathbf{R}, i = 1, \dots, n_f$.tel que :

$$\begin{aligned} J^*(x) = \min_{\zeta} f_o(\zeta, x) \\ \text{sous } f_i(\zeta, x) \leq 0, \quad i = 1, \dots, n_f \end{aligned} \quad (7)$$

La détermination de l'optimum $\zeta^*(x)$ comme une fonction du paramètre x est appelée **Programmation multi paramétrique**, la fonction à valeur réelle $J^*(x)$, qui exprime la valeur minimale de la fonction coût ou du critère en fonction de x ,est appelée **La fonction optimale**.

Définissons maintenant deux classes de programmation multi paramétrique utilisées dans notre application.

Définition (mp-LP):

Si $f_o(.,.)$ et $f_i(.,.)$ sont linéaires, $\zeta \in \mathbf{R}^{d_r}$ (avec $d_b = 0$) et $x \in \mathbf{R}^{n_r}$ le problème (6) est appelé **Programme Linéaire multi paramétrique (mp-LP)**.

Définition (mp-MILP) :

Si $f_o(.,.)$ et $f_i(.,.)$ sont linéaires, $\zeta \in \mathbf{R}^{d_r} \times \{0,1\}^{d_b}$ et $x \in \mathbf{R}^{n_r}$, le problème (7) est appelé **Programme Mixte Linéaire Entier multi paramétrique (mp-MILP)**.

Algorithmes :

Comme on le verra par l'a vu que le problème de commande optimale peut être reformulé en un programme multi paramétrique, ou ζ est la séquence d'entrée à optimiser et le paramètre x est l'état courant du système. Il existe deux approches pour la résolution d'un mp-MILP :

- ✦ mp-LP avec MILP,
- ✦ mp-LP avec la programmation dynamique (DP).

Le mp-LP avec le MILP traite directement le problème sous le formalisme MLD, en décomposant le mp-MILP en un deux sous programme mp et MILP et itérant entre les deux. Dans un premier temps une solution au problème MILP est obtenue, en traitant le vecteur d'état comme un paramètre libre, ce qui donne un vecteur d'entier réalisable qui est fixé par le mp-MILP, le réduisant à un mp-LP. Puis pour le vecteur d'entier constant, un problème mp-LP est résolu, et la fonction PWA résultante et répartie en polyèdres correspondants chacun à une région appelée région critique. Pour chaque région un nouveau sous problème MILP est résolu donnant un vecteur d'entier, en prenant la valeur courante de la fonction PWA comme une borne supérieure. Les solutions paramétriques, correspondant à deux solutions entières différentes sont comparées afin de supprimer les parties minimales. Si le sous problème MILP est faisable, alors l'algorithme se fini dans une des régions critiques, ce qui implique que la borne supérieur du courant est la solution optimale du problème.

Exécution :

Une telle commande par retour d'état peut être implémentée en temps réel, du moment que l'implémentation de l'entrée de commande est faite en deux étapes :

1. Le polyèdre doit être déterminé d'après les mesures de l'état x , une méthode rudimentaire est de passer par tout les ensembles de polyèdres afin de trouver celui correspondant aux inégalités des polyèdres visés, ou bien il est possible de construire un arbre de recherche binaire qui réduit le temps d'exécution on line, mais qui augmente la mémoire nécessaire.
2. lorsque le polyèdre adéquat a été trouvé, il ne reste plus qu'à identifier et évaluer la loi de commande affine correspondante.

Dans la plupart des cas, les polyèdres de loi de commande identiques forment un ensemble convexe, et peuvent être dans ce cas fusionnés et remplacés par leur union. Ce qui conduit à une loi de commande PWA avec moins de polyèdres et une complexité réduite. Une telle représentation est préférable car elle nécessite moins d'espace mémoire et un temps d'exécution réduit.

Bibliographie

[1] Encyclopédie Microsoft Encarta 2004

[2] techniques de l'ingénieur

[44] R. Ghosh, A. Tiwari, C. Tomlin, Automated symbolic reachability analysis; with application to Delta-Notch signaling automata. *Hybrid Systems : Computation and Control*, O. Maler, Amir Pnueli (Eds), no. 2623 in LNCS, pp 22-35, Springer, 2003.

[11][A.S. Matveev, A.V. Savkin, *Qualitative theory of hybrid dynamical systems*, Birkhäuser, 2000.

[12] A. Bemporad, P. Borodani, M. Manneli, Hybrid control of an automotive robotized gearbox for reduction of consumptions and emissions, *Hybrid Systems : Computation and Control*, O. Maler, Amir Pnueli (Eds), no. 2623 in LNCS, pp 81-96, Springer, 2003.

[13] A. Bemporad, M. Morari, Control of systems integrating logic, dynamics, and constraints, *Automatica*, 35(3):407-427, 1999.

[14] C. Tomlin, G.J. Pappas, J. Lygeros, D.N. Godbole, S. Sastry, Hybrid control models of next generation air traffic management, *Hybrid Systems IV*, no 1273 in LNCS, Springer 1997.

[15] C. Belta, J. Schug, T. Dang, V. Kumar, M. Mintz, G.J. Pappas, H. Rubin, P. Dunlop, Stability and reachability analysis of a hybrid model of luminescence in the marine bacterium *Vibrio fischeri*, *Proc. IEEE Conf. on Decision and Control*, pp 869-874, 2001.

[19] R. Valette, R. Champagnat, D. Andreu, H. Pingaud, *Modélisation des systèmes de production hybrides*, AGIS'97, colloque de recherche doctorale, Angers, France, 1997.

[20] M.S. Branicky, *Studies in hybrid systems : modeling, analysing and control*, PhD thesis, Massachusetts Institute of Technology, Cambridge, 1995.

[21] mémoire

[28] A. N. Michel and B. Hu, "Toward a stability theory of general hybrid dynamical systems," *Automatica*, vol. 35, pp. 371-384, 1999.

[29] A. S. Morse, *Control Using Logic-Based Switching*. London, U.K.: Springer-Verlag, 1997.

[30] M. Wicks, P. Peleties, and R. DeCarlo, "Switched controller synthesis for the quadratic stabilization of a pair of unstable linear systems," *Eur. J. Contr.*, vol. 4, pp. 140-147, 1998.

- [31] P. A. Peleties, "Modeling and design of interacting continuous-time/discrete event systems," Ph.D. dissertation, School of Electrical Engineering, Purdue Univ., West Lafayette, IN, Dec. 1992.
- [32] M. S. Branicky, "Multiple Lyapunov functions and other analysis tools for switched and hybrid systems," *IEEE Trans. Automat. Contr.*, vol. 43, no. 4, pp. 475–482, 1998.
- [33] P. Peleties and R. A. DeCarlo, "Asymptotic stability of m-switched systems using Lyapunov-like functions," in *Proc. 1991 American Control Conf.*, Boston, MA, 1991, pp. 1679–1684.
- [34] Z. Artstein, "Examples of stabilization with hybrid feedback," in *Hybrid Systems III*, R. Alur, T. A. Henzinger, and E. D. Sontag, Eds. New York: Springer, 1996, pp. 173–185.
- [35] T. I. Seidman, "Switching systems," UMBC, Baltimore, MD, Tech. Rep. 86-78, 1986.
- [36] D. Liberzon and A. S. Morse, "Basic problems in stability and design of switched systems," *IEEE Contr. Syst. Mag.*, vol. 19, no. 5, pp. 59–70, Oct. 1999.
- [37] M. Corless, Robust stability and controller design with quadratic Lyapunov functions, in : Variable structure and Lyapunov control (A. Zinober Ed.), Springer-Verlag, 1993.
- [38] M. S. Branicky, "Stability of switched and hybrid systems," in *Proc. 33rd IEEE Conf. Decision and Control*, Lake Buena Vista, FL, Dec. 14–16, 1994, pp. 3498–3503.
- [39] M. S. Branicky, "Studies in hybrid systems: Modeling, analysis, and control," Ph.D. dissertation, Dept. of Electrical and Computer Engineering, Massachusetts Institute of Technology, 1995.
- [40] H. Ye, A. N. Michel, and L. Hou, "Stability theory for hybrid dynamical systems," *IEEE Trans. Automat. Contr.*, vol. 43, no. 4, pp. 461–474, Apr. 1998.
- [41] A. N. Michel, "Recent trends in the stability analysis of hybrid dynamical systems," *IEEE Trans. Circuits Syst. I*, vol. 46, no. 1, pp. 120–134, Jan. 1999.
- [42] H. Ye, A. N. Michel, and L. Hou, "Stability theory for hybrid dynamical systems," in *Proc. 34th IEEE Conf. Decision and Control*, New Orleans, LA, 1995, pp. 2679–2684.
- [16] B. Hu and A. N. Michel, "Some qualitative properties of multirate digital control systems," *IEEE Trans. Automat. Contr.*, vol. 44, Apr. 1999.
- [43] R. N. Shorten and K. S. Narendra, "On the stability and existence of common Lyapunov functions for stable linear switching systems," in *Proc. 37th IEEE Conf. Decision and Control*, 1998, pp. 3723–3724.

- [44] M. A. Wicks, P. Peleties, and R. A. DeCarlo, "Construction of piecewise Lyapunov functions for stabilizing switched systems," in *Proc. 33rd IEEE Conf. Decision and Control*, Lake Buena Vista, FL, Dec. 14–16, 1994, pp. 3492–3497.
- [45] E. Feron, "Quadratic stabilizability of switched systems via state and output feedback," Center for Intelligent Control Systems, Massachusetts Institute of Technology, Tech. Rep. CICS P-468, Feb. 1996.
- [46] M. A. Wicks and R. A. DeCarlo, "Solution of coupled Lyapunov equations for the stabilization of multimodal linear systems," in *Proc. 1997 Amer. Control Conf.*, Albuquerque, NM, June 1997, pp. 1709–1713.
- [47] A. Y. Pogromsky, M. Jirstrand, and P. Spangéus, "On stability and passivity of a class of hybrid systems," in *Proc. 37th IEEE Conf. Decision and Control*, San Diego, CA, 1998, pp. 3705–3710.
- [48] L. Xie, S. Shishkin, and M. Fu, "Piecewise Lyapunov functions for robust stability of linear time-varying systems," *Syst. Contr. Lett.*, vol. 31, pp. 165–171, 1997.
- [49] S. Boyd, L. El Ghaoui, E. Feron, and V. Balakrishnan, *Linear Matrix Inequalities in System and Control Theory*. Philadelphia, PA:SIAM, 1994.
- [50] M. Johansson and A. Rantzer, "Exponential stability of hybrid systems using piecewise quadratic Lyapunov functions resulting in an LMI problem," in *Proc. 14th IFAC World Congress*, Beijing, China, July 1999.
- [51] J. Malmberg, B. Bernhardsson, and K. J. Astrom, "A stabilizing switching scheme for multi-controller systems," in *Proc. IFAC World Congress*, San Francisco, CA, July 1996.
- [52] S. Pettersson and B. Lennartson, "Stability and robustness for hybrid systems," in *Proc. 35th IEEE Conf. Decision and Control*, Kobe, Japan, 1996, pp. 1202–1207.
- [53] M. Johansson and A. Rantzer, "Computation of piecewise quadratic Lyapunov functions for hybrid systems," *IEEE Trans. Automat. Contr.*, vol. 43, no. 4, pp. 555–559, 1998.
- [54] P. Gahinet, A. Nemirovski, A. J. Laub, and M. Chilali, *LMI Control Toolbox for Use with Matlab*: Mathworks, May 1995.
- [55] L. El Ghaoui, *LMITOOL: An Interface to Solve LMI Problems*: ENSTA, 1995.
- [56] M. Johansson, "Piecewise linear control systems," Lund Institute of Technology, Lund, Sweden, ISRN LUTFD2/TFRT-1050-SE, 1999.
- [57] Drakunov, S. (1992). Sliding mode observer based on equivalent control method. *IEEE-CDC, Tucson, USA*, pp. 2368–2369.
- [58] Drakunov, S., and V. Utkin. (1995). Sliding mode observer. tutorial.

IEEE-CDC, pp. 3376–3379.

[59] Fliess M. (1990). Generalized Controller Canonical Forms for Linear and Nonlinear Dynamics. *IEEE Trans on Automatic Control*, 35, No 9, pp. 994–1000.

[60] Barbot, J.P., T. Boukhobza and M. Djemai. (1996). Sliding mode observer for triangular input form. *IEEE-CDC, Japan*.

[61] Boukhobza, T., M. Djemai, and J. P. Barbot. (1996). Nonlinear sliding observer for systems in output and output derivative injection form. *In proc. of IFAC World congress San Francisco, 1996*.

[64] Bemporad, A. and M. Morari: Control of Systems Integrating Logic, Dynamics and Constraints. *Automatica*, 35(3):407–427, March 1999.

[66] Morari, M., M. Baoti'c and F. Borrelli: Hybrid Systems Modeling and Control. *European Journal of Control*, 9(2–3):177–189, 2003

[65] Bemporad, A., F. Borrelli and M. Morari: Piecewise Linear Optimal Controllers for Hybrid Systems. In *Proceedings of the American Control Conference*, pages 1190–1194, Chicago, IL, June 2000.

[68] Borrelli, F., M. Baoti'c, A. Bemporad and M. Morari: Dynamic Programming for Constrained Optimal Control of Discrete-Time Linear Hybrid Systems. *Automatica*, 41, 2005. accepted, preprint available from <http://control.ee.ethz.ch/>.

[69a] Baoti'c, M., F.J. Christophersen and M. Morari: Infinite Time Optimal Control of Hybrid Systems with a Linear Performance Index. In *Proceedings of the 42nd IEEE Conference on Decision and Control*, pages 3191–3196, Maui, Hawaii, USA, December 2003.

[69b] Baoti'c, M., F.J. Christophersen and M. Morari: A new Algorithm for Constrained Finite Time Optimal Control of Hybrid Systems with a Linear Performance Index. In *Proceedings of the European Control Conference*, Cambridge, UK, September 2003.

[70] Mayne, D.Q., J.B. Rawlings, C.V. Rao and P.O.M. Scokaert: Constrained model predictive control: Stability and Optimality. *Automatica*, 36(6):789–814, June 2000.