

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

Université Abderrahmane Mira, Bejaia

Faculté de Technologie

Département de Génie Electrique



Mémoire de fin d'étude

En vue d'obtention du diplôme de Master en Electronique

Option : Automatique

Thème:

Simulation et commande de systèmes hybrides

Réalisé par :

Mr. Idir ZANE

Encadreur

Mr H.HADDAR

Membres de jury

Mr B.MENDIL

Mr H.LEHOUCHE

Année universitaire 2011-2012

Résumé

L'objectif de ce travail est d'apporter une solution au contrôle optimal à temps final fixé des systèmes dynamiques hybrides. L'approche utilisée est basée sur la programmation dynamique et la résolution numérique de l'équation de HJB, obtenue à partir d'un critère de performance contenant des pénalités sur l'évolution continue sur l'état final, ainsi que sur les transitions discrètes. La méthode est appliquée à un modèle de véhicule pour la commande automatique optimale de la transmission. Les équations de HJB constituent un système d'inégalités quasivariationnelles qu'il faut résoudre par une méthode numérique, une fois ces dernières résolues, c'est-à-dire que la fonction de valeur est calculée, on peut tirer le contrôle optimal à partir de la solution trouvée.

L'interaction des dynamiques continues et discrètes des SDH rend la conception d'un contrôleur efficace très délicate. Ainsi, le choix de la méthode de conception de la commande doit prendre en compte toutes les dynamiques existantes, afin de satisfaire les exigences et les contraintes fixées. Les avantages que présentent les aspects de la commande optimale ont fait d'elle un candidat idéal pour la synthèse de la contrôlabilité des SDH. Dans ce travail nous avons expliqué et utilisé la programmation dynamique car elle est bien adaptée pour la dynamique non linéaire de notre système. Son principe consiste à choisir un contrôle optimal dans un domaine fini et discrétisé basé sur l'optimisation. Les résultats obtenus sont satisfaisants mais ils dépendent du pas de discrétisation. Pour obtenir des résultats précis, il faut choisir un pas très petit, ceci conduit à des calculs prohibitifs. Nous avons utilisé un algorithme pour l'approximation de la solution optimale à commutation, car pour l'instant, il n'existe pas vraiment une méthode de résolution des SQVIs. Afin d'obtenir une commande optimale efficace, il faut rendre l'erreur d'approximation de la solution très petite voir même négligeable.

Remerciements

Je remercie dieu de m'avoir guidé et aidé tout au long de mon travail et de ma formation et surtout pour la santé et la patience qui ont été la raison de ma réussite.

Je tiens à remercier aussi **M^r H.HADDAR**, pour l'honneur qu'il m'a fait en acceptant de m'encadrer et pour ses précieux et judicieux conseils.

J'exprime ici ma gratitude à toute personne ayant contribué à ma formation.

Je remercie également les membres de jury, **Mr B.MENDIL** et **Mr H.LEHOUCHE** qui ont accepté d'examiner mon travail.

Pour finir, je remercie ma famille et mes amis. Merci à mes parents, à mon frère et à mes sœurs, pour leur patience et leur soutien.

Table des matières

Introduction générale	10
Chapitre I : Etude des systèmes dynamiques hybrides	
Introduction.....	12
I.1. Systèmes dynamiques.....	12
I.1.1 Système dynamique continu.....	12
I.1.2 Système dynamique discret.....	12
I.1.3 Système dynamique hybride.....	14
I.1.3.1. Définition	14
I.1.3.2. Différents types de modèles hybrides.....	14
I.1.3.2.1. Définition du modèle hybride de Pettersson.....	15
I.1.3.2.2 Systèmes hybrides affines considérés.....	16
I.2. Automates	17
I.2.1. Définition d'un automate à états finis.....	17
I.2.2. Définition d'un automate temporisé.....	19
I.2.3. Définition d'un automate hybride.....	20
I.3. Ensemble de temps hybride.....	23
I.4. Trajectoire et exécution d'un système dynamique hybride.....	26
I.4.1. Trajectoire hybride.....	26
I.4.2. Exécution hybride.....	26
I.4.2.1. Classification d'exécution hybride.....	27
I.5. Définition des systèmes à commutation.....	28
I.6. Classes de phénomènes hybrides.....	29
I.6.1. Commutation autonome.....	29
I.6.2. Commutation contrôlée.....	30

Table des matières

I.6.3. Sauts autonomes.....	32
I.6.4. Sauts contrôlés.....	32
Conclusion.....	32
Chapitre II : Commande optimale des systèmes dynamiques	
Introduction.....	33
II.1. Le problème de commande optimale.....	33
II.2. Existence de trajectoires optimales.....	34
II.2.1. Contrôlabilité de systèmes.....	34
II.2.1.1. Contrôlabilité des systèmes généraux.....	34
II.2.1.2. Contrôlabilité des systèmes affines.....	35
II.3. Les deux grands principes de résolution d'un problème de commande optimale.....	36
II.4. Principe du maximum de Pontryaguine.....	36
II.4.1. Enoncé général.....	36
II.5. Programmation dynamique.....	38
II.5.1. Principe d'optimalité.....	39
II.5.1.1. Solution arrière.....	39
II.5.1.2. Solution avant.....	42
II.5.2. Control optimal utilisant la programmation dynamique.....	43
II.6. Equation de Hamilton Jacobi de Bellman.....	50
II.7. Conclusion.....	56
Chapitre III : Programmation dynamique et résultats de simulation	
Introduction.....	57
III.1. Commande optimale de systèmes à commutation.....	57
III.2. Système d'Inégalités Quasi Variationnelles de la commande optimale à commutations.....	60

Table des matières

III.3. Les Inégalités Quasi Variationnelles récursives.....	63
III.4. Implémentation de l'algorithme pour la solution des QVIs.....	66
III.5. Modèle théorique du véhicule.....	69
III.6. Résultats de simulation.....	72
Conclusion.....	86
Conclusion générale.....	87

Table des figures

(1.1) Evolution continue d'un système continu.....	12
(1.2) Trajectoire d'un système à évènement discret.....	13
(1.3) Automate fini.....	18
(1.4) Automate temporisé.....	20
(1.5) Trajectoire et graphe dirigé correspondant au système de thermostat.....	23
(1.6) Ensemble de temps hybride.....	24
(1.7) Ensemble de temps hybride ($\tau \subset \tilde{\tau}, \tau \subset \hat{\tau}$).....	25
(1.8) Exemples d'ensembles de temps hybrides d'exécution finie, infinie et Zeno.....	28
(1.9) (a) Trajectoire d'une boule de billard et (b) automate associé.....	30
(1.10) Système d'embrayage mécanique.....	31
(2.1) Chemin optimal de A à B.....	39
(2.2) Processus de décision à plusieurs étages.....	40
(2.3) Processus de décision à plusieurs étages: Solution arrière.....	41
(2.4) Processus de décision à plusieurs étages.....	43
(2.5) Cadre de programmation dynamique de l'Etat optimale.....	49
(3.1) Formule de l'interpolation polynomiale parabolique.....	68
(3.2) Schéma de principe de train de propulsion.....	69
(3.3) Caractéristiques du rendement pour les différentes Vitesses, $r=[15\ 20\ 60\ 90][\text{km/h}]$...72	72
(3.4) Schéma du système sous SIMULINK.....	73
(3.5) Graphe d'état (Flow Chart) de la supervision.....	74
(3.6) Résultat obtenue en utilisant une commande avec supervision discrète (logique).....	74
(3.7) Evolution de l'état de système en boucle fermée et la commande par retour d'état obtenu par programmation dynamique de l'exemple du système continu. $\delta = 0.1$	77
(3.8) Evolution de l'état du système continu du premier ordre ainsi que la commande par retour d'état pour $\delta = 0.02$	79

Table des figures

(3.9) Caractéristiques de rendement de l'accélérateur pour chaque vitesse.....	80
(3.10) Loi de commutation obtenue avec un maillage serré ($\delta = 0.1$).....	83
(3.11) Variations des états continus $x(t)$, de l'état discret $q(t)$, de la commande $u(t)$ et de la fonction coût en fonction du temps.....	85

Liste des tableaux

(1.1) Calcul du coût au cours de la dernière étape $k=2$	47
(2.2) Calcul du coût au cours de la dernière étape $k=1$	48
(3.3) Résumé de l'approche de la procédure de Hamilton-Jacobi-Bellman.....	53

Introduction générale

Depuis toujours les systèmes dynamiques ont été conçus pour fonctionner d'une manière continue ou discrète. La modernisation de l'industrie et la croissance de la complexité des procédés, d'un côté, et l'évolution des outils de modélisation et de contrôle, de l'autre, ont conduit les chercheurs à penser à traiter les systèmes dynamiques selon leur complexité. Dans la plupart des cas, ces derniers ne sont pas purement discrets ou purement continus, mais ils intègrent les deux aspects. Ceux-ci sont généralement rencontrés parmi les systèmes mécaniques ou les systèmes continus commandés par un ordinateur.

Les systèmes dynamiques hybrides concernent plusieurs secteurs de l'industrie. Ils sont utilisés pour réguler des systèmes complexes présentant des évolutions continues dans le temps et des phénomènes discrets qui interagissent entre eux, par exemple : l'automobile (injection), avionique (commande de vol), le nucléaire (contrôle du système de refroidissement)...etc. Les fonctionnements de ces derniers doivent être fiables et précis pour bien accomplir leurs tâches. En plus, la plupart des systèmes complexes sont difficiles à commander par des procédés seulement continus ou discrets. C'est pourquoi des méthodes d'analyse et de conception de modèles hybrides ont été créées et développées.

Un système dynamique hybride est composé de deux parties (partie continue, partie discrète). Les dynamiques continues sont modélisées par des équations différentielles, par contre les phénomènes discrets par un automate fini. Le modèle obtenu est appelé alors automate hybride.

La théorie de la commande analyse les propriétés des systèmes dynamiques sur lesquels on peut agir au moyen d'un contrôle sur le système. Le but est alors d'amener le système d'un état initial donné à un certain état final, en respectant éventuellement certaines contraintes. Les systèmes abordés peuvent être : systèmes différentiels, systèmes discrets, systèmes avec bruit...etc. Leurs origines sont très diverses : mécanique, électrique, électronique, ...etc. L'objectif peut être de stabiliser le système pour le rendre insensible à certaines perturbations (stabilisation) ou encore de déterminer des solutions optimales pour un certain critère d'optimisation (contrôle optimal, ou commande optimale) [9].

L'objectif de ce travail est d'apporter une solution au contrôle optimal à temps final fixé des systèmes dynamiques hybrides. L'approche utilisée est basée sur la programmation

Introduction générale

dynamique et la résolution numérique de l'équation de HJB, obtenue à partir d'un critère de performance contenant des pénalités sur l'évolution continue sur l'état final, ainsi que sur les transitions discrètes. La méthode est appliquée à un modèle de véhicule pour la commande automatique optimale de la transmission.

Nous commençons notre mémoire par des généralités et définitions concernant les systèmes hybrides ainsi que la méthode utilisée pour modéliser leur comportement. Dans le 2^{ème} chapitre, nous donnons une étude de la commande optimale des systèmes dynamiques, à savoir le principe de Pontryaguine et la programmation dynamique. Dans le troisième chapitre, nous introduisons l'algorithme implémenté ainsi que le modèle dynamique d'un véhicule à quatre vitesses. Le modèle dynamique adopté pour le véhicule a été analysé sous MATLAB en utilisant SIMULINK et STATE FLOW. Le programme établi a été testé sur un exemple théorique ensuite appliqué au modèle du véhicule.

Introduction

Les systèmes dynamiques hybrides sont des systèmes qui combinent deux types de dynamiques (dynamiques continues et dynamiques discrètes). Ils concernent les systèmes complexes qui ne peuvent pas être modélisés par des dynamiques seulement continues ou discrètes.

I.1 Systèmes dynamiques

I.1.1 Système dynamique continu [1]

Des grandeurs telles que la position, la vitesse, l'accélération, le niveau, la pression, la température, le débit, la tension, le courant...etc, sont des variables continues, dans le sens qu'elles peuvent prendre n'importe quelle valeur dans l'ensemble \mathbb{R} lorsque le temps, lui-même est continu. Un système dynamique continu est généralement décrit par un système d'équations différentielles du type:

$$\dot{x} = f(x(t), q, u(t)) \quad (1.1)$$

$X \in \mathbb{R}^n$ est l'état continu du système, f est une fonction non linéaire de l'état.

On sait que le comportement d'un système continu à l'instant t peut être résumé par son état qui représente la mémoire minimale du passé nécessaire à la détermination du futur.

La figure (1.1) représente un exemple d'évolution continue d'un système à deux états (x_1, x_2).

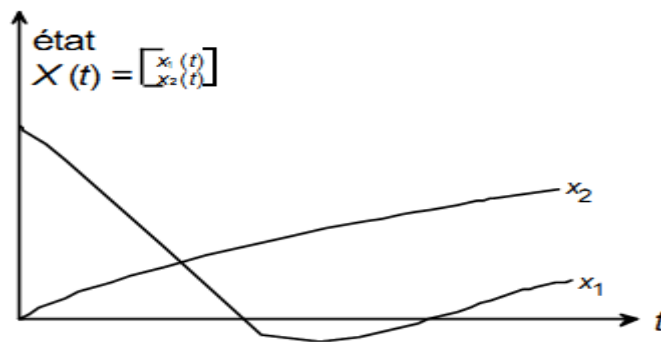
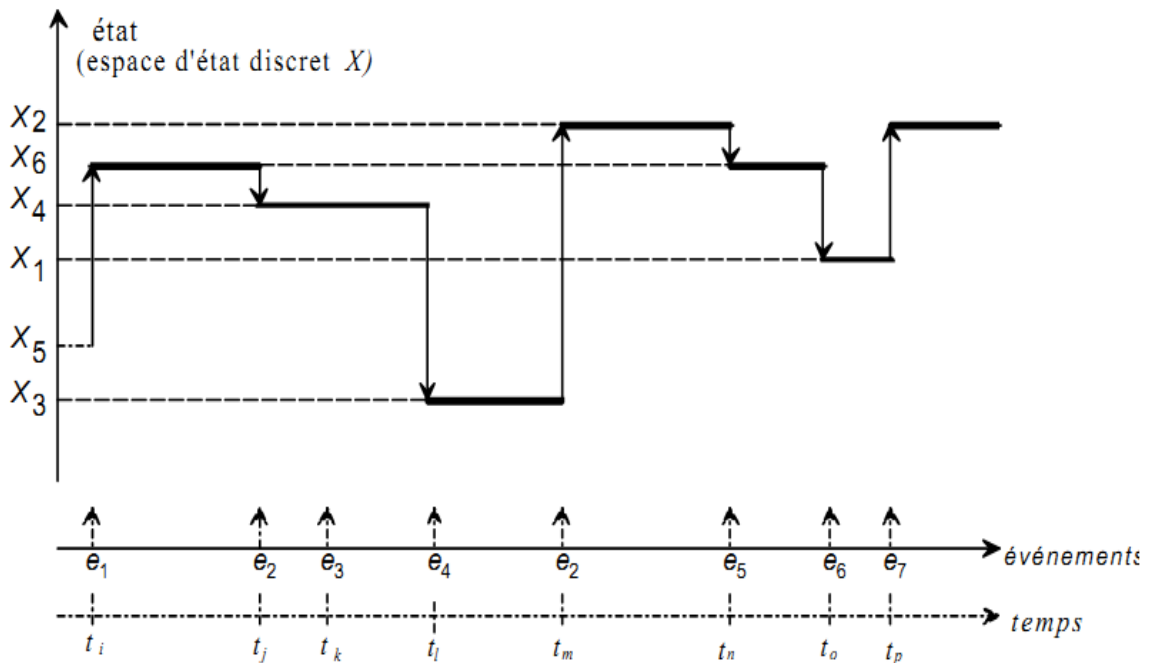


Figure (1.1) Evolution continue d'un système continu

I.1.2 Système dynamique discret [1]

Les systèmes automatisés de production, à l'initiative de l'homme, sont caractérisés par une forte complexité et flexibilité, selon un certain point de vue, ils peuvent être spécifiés par des modèles à événements discrets, tel que les systèmes de trafic (aérien, ferroviaire,...), les systèmes de communication...etc, sont d'autres exemples de systèmes dynamiques dont l'activité est due aux occurrences asynchrones d'événements discrets. Les grandeurs auxquelles nous avons affaire sont des constantes, telles que le nombre de produits dans un stock, le nombre de processeurs en activité...etc.

La figure (1.2) montre un exemple d'une trajectoire d'un système à évènement discret [1]:



Figure(1.2) trajectoire d'un système à évènement discret

$X = \{X1, X2, X3, X4, X5, X6\}$ ensemble (discret) d'états= espace d'états

$\Sigma = \{e1, e2, e3, e4, e5, e6, e7\}$ ensemble d'évènement discret

La trajectoire est constante par morceaux, elle saute d'un état à un autre avec l'occurrence d'un évènement.

Exemple :

- un même évènement (e2) peut conduire à des états différents
- des évènements différents (e1 et e5) peuvent conduire à un même état
- des évènements (e3) peuvent se produire et être inactifs pour le système
- la séquence des états et le temps de maintien associé à chaque état caractérisent la trajectoire

I.1.3 Système dynamique hybride

I.1.3.1 Définition[17]

Les systèmes réels sont souvent des systèmes complexes dont la dynamique est modélisée, d'un point de vue macroscopique, par des phénomènes discrets et continus.

Il apparaît alors évident que ces systèmes ne peuvent se contenter d'une représentation homogène à dynamique purement continue ou purement événementielle. Des études visant à considérer les interactions entre ces deux mondes s'imposent.

Dans ce contexte, une attention particulière s'est portée sur les systèmes dynamiques hybrides ces dernières années et de nombreuses approches de modélisation traitant à la fois les aspects continus et discrets ont été proposées. Les formalismes qui en découlent présentent des propriétés plus ou moins intéressantes en termes de modularité, réutilisabilité, structuration des données.

La modélisation des systèmes dynamiques hybrides doit décrire deux comportements: d'une part, la dynamique continue représentée par un système d'équations différentielles et algébriques et d'autre part, la dynamique discrète représentée par un ensemble d'états et de transitions.

I.1.3.2 Différents types de modèles hybrides [3]

Dans la littérature, il existe une multitude de définitions de modèles dynamiques hybrides. Ces modèles sont plus ou moins généralistes et plus ou moins adaptés à la simulation ou à l'analyse. Le modèle doit représenter mathématiquement le comportement et les interactions entre les parties continues et discrètes du système.

Certains modèles sont basés sur une représentation d'un système continu supervisé par un réseau de Pétri dont les transitions sont influencées par l'état continu.

D'autres modèles utilisent des transformations interfaces pour établir une relation entre le vecteur d'état réel et des événements. Ces événements animent un automate dont l'état discret agit à son tour sur le système continu.

Beaucoup de modèles définissent des ensembles de commutations dans l'espace de phase. Un événement est généré lorsque l'état continu pénètre dans une de ces régions de commutation. Le système discret stimulé par ces événements agit sur le système continu en le faisant commuter.

Avec ce genre de modèle, l'état discret peut aussi dépendre d'événements extérieurs en entrée du système. Le modèle le plus généraliste est obtenu en ajoutant des sauts de valeurs dans les variables d'état continues lors des commutations du système discret.

Nous utilisons une représentation hybride simple, à base de systèmes continus commutés dans certaines régions. Ce type de modélisation est proposé par Patterson.

1.1.3.2.1 Définition du modèle hybride de Patterson[3]

La définition du modèle dynamique hybride, présentée ici, est assez générale et convient à une analyse de stabilité. La caractéristique essentielle d'un système hybride est de contenir à la fois un vecteur d'état continu $X(t)$ à valeurs réelles, et un vecteur d'état discret $E(t)$ dont les valeurs appartiennent à un ensemble discret fini M .

Le terme système dynamique hybride vient de la coexistence de ces deux types de variables. Le système continu est modélisé par une équation différentielle ordinaire, ayant la particularité de dépendre du vecteur d'état discret $E(t)$. Le système discret est un système événementiel dont les événements, provoquant un changement d'état discret, sont issus soit d'une entrée discrète, soit d'un événement lié à l'état du système continu.

Dans cette étude, ces derniers événements sont liés à la traversée par le vecteur d'état de surfaces du plan de phase dites surfaces de commutations.

Bien qu'un système hybride puisse être aussi bien à temps continu qu'à temps discret, nous limitons la définition suivante aux systèmes à temps continu.

Définition I.1 : Un système hybride de Patterson $H = (R^n \times M, R^p \times \Sigma, F, \emptyset)$ est défini par:

- un ensemble non-vide appelé l'espace d'états hybride $H = R^n \times M$;

un ensemble $R^p \times \Sigma$ appelé l'espace des entrées externes de H , des fonctions de transition $F : D_F \rightarrow R^n$ et $\emptyset : D_\emptyset \rightarrow M$ vérifiant

$$\begin{cases} \dot{X}(t) = F(X(t), E(t), u(t)) \\ E^+(t) = \emptyset(X(t), E(t), \sigma(t)) \end{cases} \tag{1.2}$$

et

$$\begin{cases} D_F \subseteq R^n \times M \times R^p \\ D_\emptyset \subseteq R^n \times M \times R^p \times \Sigma \end{cases} \tag{1.3}$$

Où $E^+(t)$ désigne l'état $E(t)$ immédiatement après l'instant t .

La composante X , qui est à valeurs réelles, du vecteur d'état du système hybride H est appelée variable d'état continue. De même, la composante E , qui est à valeurs discrètes, du vecteur d'état est appelée variable d'état discrète. De manière identique pour les entrées du système, on appelle chaque élément u une entrée continue du système et chaque élément σ une entrée discrète.

La transition d'un état discret à un autre se produit lorsqu'un événement est détecté. Il ya deux types d'évènements : les évènements d'états et les évènements temporels. Les évènements d'états surviennent quand l'état $X(\cdot)$ atteint un ensemble de commutation $S_{E_i E_j}$ qui représente l'hypersurface de transition de l'état discret $E(t) = i$ à un autre $E^+(t) = j$, alors que les évènements temporels sont des évènements qui se réalisent quand des dates prédéfinis sont atteintes.

I.1.3.2.2 Systèmes hybrides affines considérés [3]

Un système hybride est dit affine par morceaux si la fonction $F(\cdot, \cdot)$ est définie pour chaque $E \in M$ et $X \in \mathbb{R}^n$ par $F(X(t), E(t)) = A_{E(t)}X + B_{E(t)}$, où $A_{E(t)} \in \mathbb{R}^{n \times n}$ $B_{E(t)} \in \mathbb{R}^n$ sont des matrices qui dépendent de l'état discret $E(t)$. Ainsi, un système hybride affine par morceaux est un système hybride qui est sous la forme:

$$\begin{cases} \dot{X}(t) = A_{E(t)}X + B_{E(t)} \\ E(t) \in M = \{1, 2, \dots, N\} \end{cases} \quad (1.4)$$

Remarque I.1

Le couple de fonctions (X, E) est dit solution du système hybride affine par morceaux s'il satisfait (1.4), où $E(\cdot)$ est une fonction constante par morceaux.

- Un temps t est dit temps de commutation d'une solution (X, E) de (1.4) si la fonction E est discontinue en t .

On dit qu'une trajectoire de (1.4) commute en un temps t si ce dernier est un temps de commutation.

- Si une solution (X, E) de (1.4) est définie sur $[0, +\infty[$, les temps de commutations forment une suite $\{t\}_{n=0}^{\infty}$ de nombres réels positifs telle que

$$t_0 = 0, t_{n+1} > t_n, n = \{0, 1, 2, \dots, n\} \text{ et } \lim_{n \rightarrow \infty} t_n = \infty,$$

Qu'on appelle suite de commutations de la solution (X, E) .

Les transitions d'état discrètes s'effectuent lors d'évènements de trois sortes:

Evénements d'état ou frontière fixe: Dans l'espace d'état, les commutations se produisent en des surfaces de commutations qui se composent d'hyperplans de dimension $n - 1$:

$$S_{EiEj} = \{X(t) \in \mathbb{R}^n : N_{ij} \cdot X \leq l_{ij}\} \quad (1.5)$$

Avec $N_{ij} \in \mathbb{R}^{1 \times n}$ et $l_{ij} \in \mathbb{R}$.

Événements périodiques: D'une manière générale, pour les systèmes dynamiques hybrides non autonomes, la fonction \emptyset peut dépendre aussi du temps $\emptyset(X, E, t) : \mathbb{R}^n \times X \times M \times \mathbb{R} \rightarrow M$. Dans ce cas, des événements temporels peuvent avoir lieu et valider une transition vers un autre état discret. Ces événements peuvent être, par exemple, des événements périodiques. Ces derniers sont simplement définis par les instants $t = dP_{EiEj}$ où dP_{EiEj} appartient à l'ensemble :

$$P_{EiEj} = \{t \in \mathbb{R} : t = kT + \varphi, k \in \mathbb{N}\} \quad (1.6)$$

avec T la période et φ la phase de ces événements périodiques.

Frontière mobile: Les commutations peuvent se réaliser aussi en des surfaces obliques de commutations définies par:

$$S_{EiEj} = \{X(t) \in \mathbb{R}^n, \exists t^* \in \mathbb{R} : N_{ij} \cdot X \leq \Delta_{ij} \cdot t + b_{ij} \forall t < t^*\} \quad (1.7)$$

t^* vérifie $N_{ij} \cdot X(t^*) = \Delta_{ij} \cdot t^* + b_{ij}$, où $N_{ij} \in \mathbb{R}^{1 \times n}$, $\Delta_{ij} \in \mathbb{R}$ et $B_{ij} \in \mathbb{R}$.

I.2 Les automates

I.2.1 Définition d'un automate à états finis [4]

Définition I.2 Un automate fini est défini par un 5-uplet $A = (\Sigma, Q, q_0, F, \delta)$ où :

- Σ est un ensemble fini de symboles (l'alphabet)
- Q est un ensemble fini d'états
- $q_0 \in Q$ est l'état initial
- $F \subset Q$ sont les états finaux
- δ est une fonction totale de $(Q \times \Sigma \rightarrow Q)$, appelée fonction de transition.

Chapitre I

Etude des systèmes dynamiques hybrides

Un automate fini correspond à un graphe orienté, dans lequel certains des nœuds (états) sont distingués et marqués comme initiaux ou finaux et dans lequel les arcs (transitions) sont étiquetés par des symboles de Σ . Si $\delta(q, a) = r$, on dit que a est l'étiquette de la transition (q, r) . Les automates admettent une représentation graphique, comme celle de la figure(1.3). Dans cette représentation, l'état initial 0 est marqué par un arc entrant sans origine et les états finaux (ici l'unique état final est 2) par un arc sortant sans destination. La fonction de transition correspondant à ce graphe s'exprime matriciellement par :

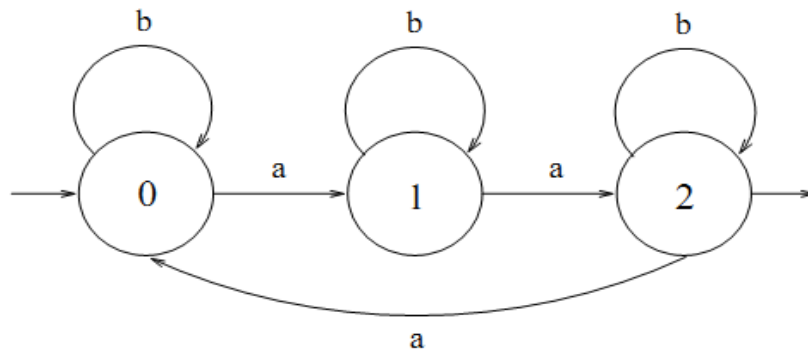


Figure (1.3) Automate fini

δ	a	b
0	1	0
1	2	1
2	0	2

Un calcul dans A est une séquence d'états $q_1 \dots q_n$ de A , tel qu'il existe toujours au moins une transition entre deux états successifs q_i et q_{i+1} . L'étiquette du calcul est le mot construit par concaténation des étiquettes de chacune des transitions. Un calcul dans A est réussi si l'état d'origine est l'état initial, et si l'état terminal est l'un des états finaux. Le langage reconnu par l'automate A , noté $L(A)$, est l'ensemble des étiquettes des calculs réussis. Dans l'exemple précédent, le mot $baab$ appartient au langage reconnu, puisqu'il étiquette le calcul : 00122 .

La notation \vdash_A permet de formaliser cette notion. Ainsi on écrira, pour a dans Σ et v dans Σ^* :

$(q, av) \vdash_A (\delta(q, a), v)$ pour noter une étape de calcul. Cette notation s'étend en

$(q, uv) \vdash_A^* (p, v)$ s'il existe une suite d'états

$q = q_1 \dots q_n = p$ tels que $(q_1, u_1 \dots u_n v) \vdash_A (q_2, u_2 \dots u_n v) \dots \vdash_A (q_n, v)$. Avec ces

notations, on a :

$$L(A) = \{u \in \Sigma^* \mid (q_0, u) \vdash_A^* (q, \varepsilon), \text{ avec } q \in F\} \quad (1.8)$$

Cette notation met en évidence l'automate comme une machine permettant de reconnaître des mots: tout parcours partant de q permet de «consommer» un à un les symboles du mot à reconnaître, ce processus stoppe lorsque le mot est entièrement consommé : si l'état ainsi atteint est un état final, alors le mot appartient au langage reconnu par l'automate. On dérive un algorithme permettant de tester si un mot appartient au langage reconnu par un automate fini.

La complexité de cet algorithme découle de l'observation que chaque étape de calcul correspond à une application de la fonction $\delta()$, qui, elle-même, se réduit à la lecture d'une case d'un tableau et une affectation, deux opérations qui s'exécutent en temps constant. La reconnaissance d'un mot u se calcule en exactement $|u|$ étapes.

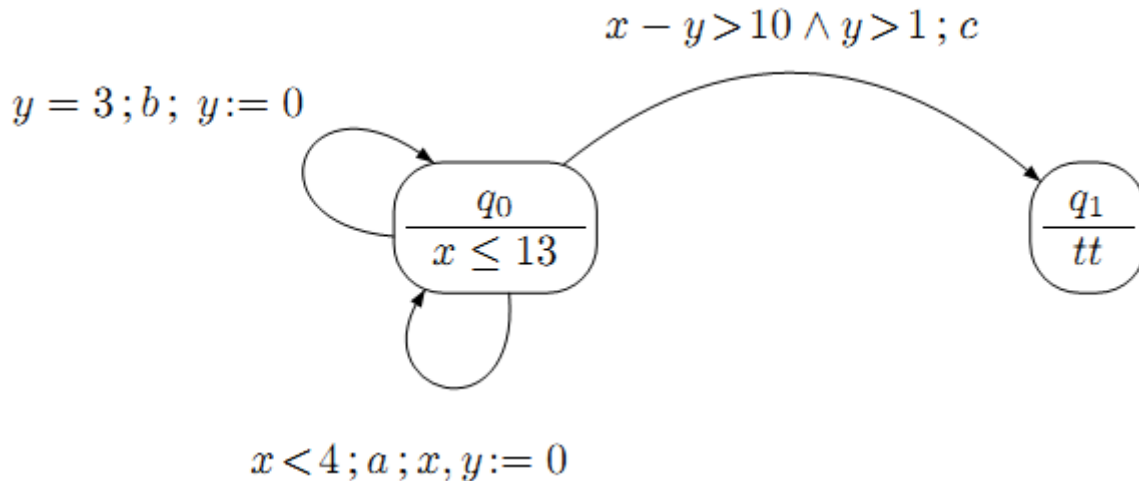
I.2.2 Définition d'un automate temporisé [5]

Les automates temporisés ont été introduits par R. Alur et D. Dill dans les années 90. Il s'agit d'automates classiques munis d'horloges qui évoluent de manière continue avec le temps. Chaque transition contient une garde (sur la valeur des horloges) décrivant quand la transition peut être exécutée et un ensemble d'horloges qui doivent être remises à zéro lors du franchissement de la transition. Chaque état de contrôle contient un invariant (une contrainte sur les horloges) qui peut restreindre le temps d'attente dans l'état et donc forcer l'exécution d'une transition d'action. En général, le domaine de temps peut être \mathbb{N} ou \mathbb{R}^+ .

Un automate temporisé est un 6-uplet $(Q, q_0, X, \text{Inv}, T, \Sigma)$ avec :

- Q est un ensemble fini d'états de contrôle
- $q_0 \in Q$ est l'état initial
- X est un ensemble fini d'horloges (à valeur réelle positive)
- $\text{Inv} : Q \rightarrow \mathcal{C}(V)$ associe un invariant à chaque état de contrôle.
- $T \subseteq Q \times \mathcal{C}(X) \times \Sigma \times 2^X \times Q$ est un ensemble fini de transitions $e(q, g, a, r, q') \in T$ représente une transition de q vers q' , g est la garde associée à e , r est l'ensemble d'horloge devant être remises à zéro et a est l'étiquette de e , on note $q \rightarrow q'$
- Σ est un alphabet d'action.

Un exemple d'automate temporisé est donné par la figure(1.4). Cet automate a pour localité initiale q_0 . L'invariant $x \leq 13$ de q_0 indique qu'on peut rester dans q_0 tant que x est inférieur à 13 mais pas au delà. La transition a est exécutable lorsque $x < 4$ et son franchissement remet x et y à zéro.



Figure(1.4) Automate temporisé

Un état ou une configuration d'un automate temporisé est une paire $(q, v) \in Q \times \mathbb{R}^X$ où q représente l'état de contrôle courant et v une valuation pour les horloges.

La sémantique d'un automate temporisé est définie sous la forme d'un système de transition temporisé qui comporte des transitions d'action (étiquetées par un élément de Σ) et des transitions de temps (étiquetées par des durées réelles) :

I.2.3 Définition d'un automate hybride [2]

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

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

Définition I.3 Un automate hybride H est une collection [6]:

$H = (Q, X, f, \text{Init}, D, E, G, R)$ où :

- $Q = \{q_1, q_2, \dots\}$ est un ensemble d'états discrets;
- $X = \mathbb{R}^n$ est un ensemble d'états continu;
- $f(\cdot, \cdot): Q \times X \rightarrow \mathbb{R}^n$ est un champ de vecteurs;
- $\text{Init} \subseteq Q \times X$ est un ensemble d'état initiaux;
- $\text{Dom}(\cdot): Q \rightarrow P(X)$ est un domaine ;
- $E \subseteq Q \times Q$ est un ensemble flancs
- $G(\cdot): E \rightarrow P(X)$ est une condition de garde;
- $R(\cdot, \cdot): E \times X \rightarrow P(X)$ est une fonction de reset (initialisation).

Avec $P(X)$ dénotons l'ensemble de partition de l'ensemble X . la notation de la définition I.3 suggère par exemple, que la fonction Dom assigne un ensemble d'états continus $\text{Dom}(q) \subseteq \mathbb{R}^n$ à chaque état discret $q \in Q$, par $(q, x) \in Q \times X$ on fera référence à un état de H . Les automates hybrides définissent les évolutions possibles de leur état. Ainsi partant d'un état initial $(q_0, x_0) \in \text{Init}$, l'état continu évolue selon l'équation différentielle :

$$\dot{x} = f(q_0, x), \quad x(0) = x_0 \quad (1.9)$$

Pendant que l'état discret q reste constant :

$$q(t) = q_0 \quad (1.10)$$

L'évolution continue peut continuer tant que x reste dans $\text{Dom}(q_0)$. Si à un point l'état continu atteint la condition de garde $G(q_0, q_1) \subseteq \mathbb{R}^n$ d'un flanc (limite) $(q_0, q_1) \in E$, l'état discret peut changer de valeur à q_1 . En même temps l'état continu sera initialisé à une valeur $R(q_0, q_1, x) \subseteq \mathbb{R}^n$. Après cette transition discrète, l'évolution continue et l'opération se répète.

Chapitre I

Etude des systèmes dynamiques hybrides

Pour faciliter la discussion, on supposera que le nombre d'états discrets est fini, et que pour tout $q \in Q$, le champ de vecteur $f(q, \cdot)$ est Lipschitz continue. Rappelons que ceci garantit que les solutions de l'équation différentielle $\dot{x} = f(q, x)$ sont bien définies.

Finalement nous supposons que pour tout $e \in E$, $G(e) \neq \{\} = \emptyset$, et pour tout $x \in G(e)$, $R(e, x) = \emptyset$. Cette supposition élimine des cas pathologiques (anormaux ou singuliers).

Il est souvent convenable de visualiser un automate hybride comme un graphe dirigé (Q, E) avec des nœuds Q et des flancs E . avec chaque nœud $q \in Q$, on associe un ensemble d'états initiaux $\{x \in X \mid (q, x) \in \text{Init}\}$, un champ de vecteur $f(q, \cdot): \mathbb{R}^n \rightarrow \mathbb{R}^n$ et un domaine $\text{Dom}(q) \subseteq \mathbb{R}^n$. un flanc débute à $q \in Q$ et se termine à $q' \in Q$. Avec chaque flanc $(q, q') \in E$, on associe une condition de garde $G(q, q') : \mathbb{R}^n \rightarrow \mathbb{P}(\mathbb{R}^n)$.

Les graphes dirigés contiennent exactement les mêmes informations que la définition I.3 Ils peuvent alors être considérés comme des définitions informelles des automates hybrides.

Exemple [6]

Considérons une chambre chauffée par un radiateur commandé par un thermostat. Supposons que lorsque le Radiateur est à l'état OFF, la température de la chambre ($x \in \mathbb{R}$), décroît exponentiellement vers 0° selon l'équation différentielle suivante :

$$\dot{x} = -ax, \quad a > 0 \quad (1.11)$$

Lorsque le thermostat allume le radiateur, la température de la chambre croît exponentiellement vers 30° , selon l'équation différentielle :

$$\dot{x} = -a(x - 30) \quad (1.12)$$

Supposons que le thermostat essaye de garder la température aux alentours de 20° . Pour éviter les oscillations (c-à-d faire commuter le radiateur ON et OFF tout le temps), le thermostat n'essaye pas d'allumer le radiateur jusqu'à ce que la température chute en dessous de 19° .

En raison de l'incertitude dans la dynamique du radiateur ce dernier ne s'allume effectivement que quand la température atteint 18° , de la même façon le thermostat n'essaye pas d'éteindre le radiateur que quand la température de la chambre augmente au dessus de

21°, et à cause d'incertitude dans la dynamique du radiateur, la température peut augmenter encore un peu plus jusqu'à 22° avant que la chambre commence à refroidir.

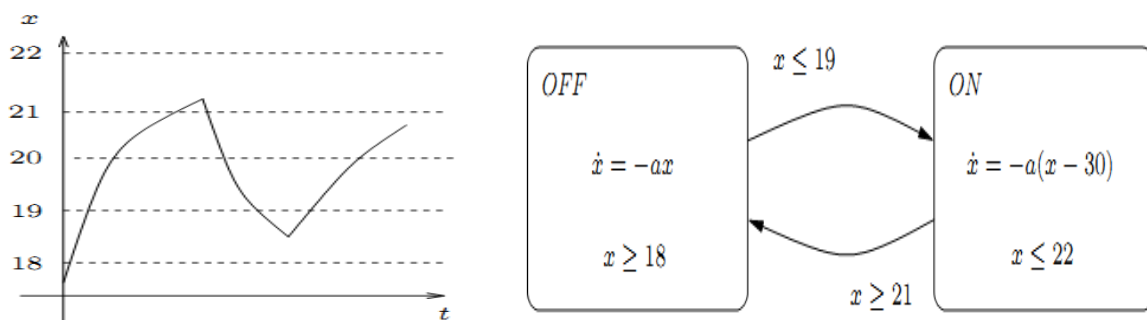
Une trajectoire du système du thermostat est représentée par la figure (1.5) notons que dans ce cas, multiple trajectoires peuvent être obtenues pour la même condition initiale, vu que pour certaines valeurs de température on a le choix d'allumer ou d'éteindre le radiateur. Les systèmes pour lesquelles un tels choix existe sont dits non déterministes. Notons que ce système a un état continu et un état discret.

L'état continu est la température $x \in \mathbb{R}$. L'état discret est $q \in \{ON, OFF\}$ informe si le radiateur est ON ou OFF. L'évolution de x est gouvernée par des équations différentielles, tandis que l'évolution de q est gouvernée à travers des sauts.

L'évolution des deux types d'état est complète. Quand $q = OFF$, alors que x décroît exponentiellement selon l'équation différentielle (1.11). Si par contre $q = ON$, x augmente selon l'équation (1.12).

De même q ne peut pas sauter de l'état ON à l'état OFF que si $x \geq 21^\circ$; q doit sauter de ON à OFF si $x \geq 22$ etc.

Il est profitable (avantageux) de décrire complètement un tel système hybride en mélangeant l'équation différentielle avec le graphe dirigé de la figure (1.5).



Figure(1.5) (a) trajectoire du système de thermostat, (b) graphe dirigé
Correspondant au système de thermostat

I.3 Ensemble de temps hybride [6]

Définition I.4

Un ensemble de temps hybride est une séquence d'intervalles $\tau = \{I_0, I_1, \dots, I_N\} = \{I_i\}_{i=0}^N$ finie ou infinie (c à d que $N = \infty$ est permis) telle que :

- $I_i = [\tau_i, \tau'_i]$ pour tout $i < N$;

- Si $N < \infty$ alors soit $I_N = [\tau_N, \tau'_N]$, soit $I_N = [\tau_N, \tau_N)$; et
- $\tau_i \leq \tau'_i = \tau_{i+1}$ pour tout i .

Un exemple d'un ensemble de temps hybride est celui donné par la figure (1.6) . Noter que le point de temps de la fin de chaque intervalle I_i correspond au point de début de l'intervalle suivant I_{i+1} (voir les points de temps t_2, t_3 de la figure (1.6)). L'interprétation est que ces instants sont les points de temps auxquels les états et τ_{i+1} correspondent juste après les transitions. Les transitions discrètes sont supposés être instantanées, ainsi $\tau'_i = \tau_{i+1}$.

L'avantage de cette convention est qu'elle permet de modéliser les situations où plusieurs transitions discrètes ont lieu l'une après l'autre à un même instant. Et dans ce cas $\tau'_{i-1} = \tau_i = \tau'_i = \tau_{i+1}$ (voir l'instant t_5 dans la figure (1.6)).

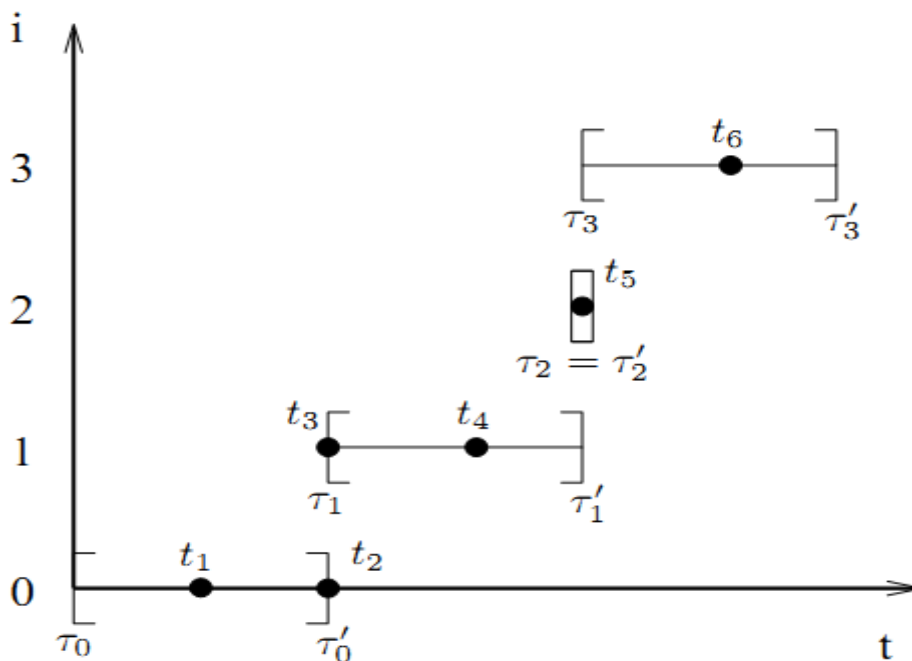


Figure (1.6) un ensemble de temps hybride

Pour $t_1 \in [\tau_i, \tau'_i] \in \tau$ et $t_2 \in [\tau_j, \tau'_j] \in \tau$, on dira que t_1 précède t_2 (dénnoté par $t_1 < t_2$) si $t_1 < t_2$ ou si $i < j$ c-à-d t_1 appartient à un intervalle qui précède celui de t_2 . Sur la figure (1.6) on a $t_1 < t_2 \leq t_3 < t_4 < t_5 < t_6$. En général, étant données deux différents instants de temps t_1 et t_2 appartenant à τ , on a soit $t_1 < t_2$ ou $t_2 < t_1$. Si on utilise la terminologie mathématique on dira que chaque ensemble de temps hybride est linéairement ordonné par la relation $<$.

Etant donné deux ensembles de temps hybride τ et $\hat{\tau}$ il y a une façon naturelle de définir si l'un est plus court que l'autre (τ est dit préfixe de $\hat{\tau}$ s'il est plus court). De manière plus formelle nous disons que $\tau = \{I_i\}_{i=0}^N$ est un préfixe de $\hat{\tau} = \{\hat{I}_i\}_{i=0}^M$ (et on écrit $\tau \subseteq \hat{\tau}$) s'ils sont soit identiques, ou τ est fini, $N \leq M$ (noter que M peut être indéfinie), $I_i = \hat{I}_i$, pour tout $i = 0, 1, \dots, N - 1$, et $I_N \subseteq \hat{I}_N$.

Pour les cas de la figure (1.7), nous dirons que τ est un préfixe stricte de $\hat{\tau}$ et de $\tilde{\tau}$, mais $\hat{\tau}$ n'est pas un préfixe de $\tilde{\tau}$ et $\tilde{\tau}$ n'est pas un préfixe de $\hat{\tau}$. notons qu'étant donné τ et $\hat{\tau}$, on peut avoir le cas où ni $\tau \subseteq \hat{\tau}$ ni $\hat{\tau} \subseteq \tau$ (comme : étant donnée deux sous ensembles de nombre réels $A \subseteq \mathbb{R}; B \subseteq \mathbb{R}$ mais ni $A \subseteq B$ ni $B \subseteq A$). Si on utilise la terminologie mathématique on peut dire que l'ensemble de tous les éléments de temps hybride est partiellement ordonné par la relation \subseteq .

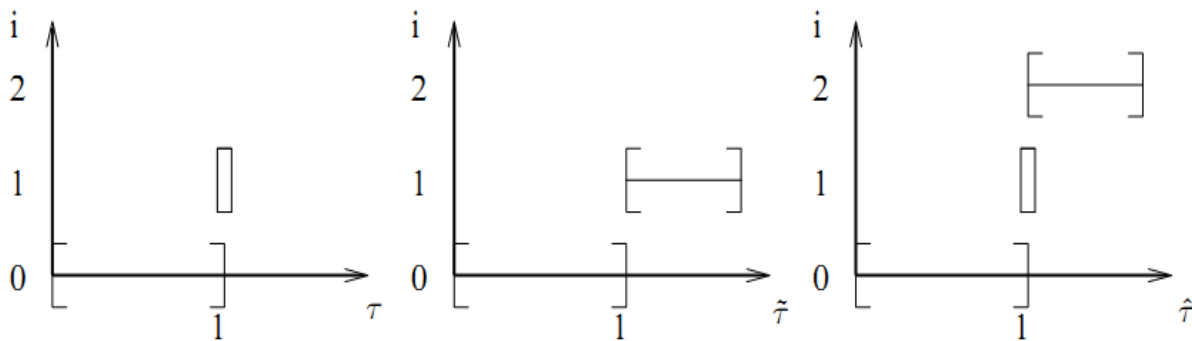


Figure (1.7) $\tau \subset \tilde{\tau}, \tau \subset \hat{\tau}$

L'ensemble de temps hybride sera utilisé pour définir l'horizon temporel dans lequel les états du système évoluent. Quelle est la signification pour un état d'évoluer dans un ensemble de temps hybride ? pour les systèmes continus avec un état $x \in \mathbb{R}^n$ une telle évolution est une fonction $x(\cdot) : [0, T] \rightarrow \mathbb{R}^n$ qui associe aux éléments de l'intervalle $[0, T]$ des éléments de \mathbb{R}^n . Pour les systèmes discrets dont les états prennent des valeurs dans un ensemble finie $q \in \{q_1, q_2, \dots, q_n\}$ une telle évolution est une séquence d'états. Pour les systèmes hybrides, où l'état a une composante continue $x \in \mathbb{R}^n$ et une composante discrète $q \in \{q_1, q_2, \dots, q_n\}$ on doit sortir avec un mélange de ces deux notions.

I.4 Trajectoire et exécution d'un système dynamique hybride [6]

I.4.1 Trajectoire hybride

Définition I.5

Une trajectoire hybride est un triple (τ, q, x) consistant en un ensemble de temps hybride $\tau = \{I_i\}_{i=0}^N$ et deux séquence de fonctions $q = \{q_i(\cdot)\}_{i=0}^N$ et $x = \{x_i(\cdot)\}_{i=0}^N$ avec $q_i(\cdot): I_i \rightarrow Q$ et $x_i(\cdot): I_i \rightarrow \mathbb{R}^n$.

Une exécution d'un automate hybride autonome est une trajectoire hybride (τ, q, x) de ses variables d'état, les éléments listés dans cette définition imposent des restrictions sur le type des trajectoires hybrides que l'automate hybride trouve acceptable.

I.4.2 Exécution hybride

Définition I.6

Une exécution d'un automate hybride H est une trajectoire hybride (τ, q, x) qui satisfait les conditions suivantes :

- Condition initiale $(q_0(0), x_0(0)) \in \text{Init}$.
- Evolution discrète : pour tout i , $(q_i(\tau'_i), q_{i+1}(\tau_{i+1})) \in E$,
 $x_i(\tau'_i) \in G(q_i(\tau'_i), q_{i+1}(\tau_{i+1}))$, et $x_{i+1}(\tau_{i+1}) \in R(q_i(\tau'_i), q_{i+1}(\tau_{i+1}), x_i(\tau'_i))$
- Evolution continue : pour tout i ,
 1. $q_i(\cdot): I_i \rightarrow Q$ est constante dans l'intervalle I_i c.-à-d $q_i(t) = q_i(\tau_i)$ pour tout $t \in I_i$
 2. $x_i(\cdot): I_i \rightarrow X$ est la solution à l'équation différentielle $\frac{dx_i}{dt} = f(q_i(t), x_i(t))$ sur I_i partant de $x_i(\tau_i)$ et
 3. Pour tout $t \in [\tau_i, \tau'_i)$, $x_i(t) \in \text{Dom}(q_i(t))$

Cette définition spécifie quelles sont parmi les trajectoires hybrides celles qui sont une exécution de H et celles qui ne le sont pas, en imposant un nombre de restrictions. La première restriction dicte qu'une exécution doit débiter à partir d'un état initial acceptable dans Init .

Pour la simplicité on utilisera $(q_0, x_0) = (q(\tau_0), x(\tau_0)) \in \text{Init}$ pour dénoter l'état initial d'une exécution (τ, q, x) . Comme pour les systèmes continus on peut supposer que $\tau_0 = 0$ sans perte de généralité. La seconde restriction détermine quand est ce que les transitions discrètes peuvent avoir lieu et qu'elle sera la valeur que l'état peut prendre après la transition discrète.

Les exigences relient l'état avant la transition $(q_i(\tau'_i), x(\tau'_i))$ à l'état après la transition $(q_{i+1}(\tau_{i+1}), x_{i+1}(\tau_{i+1}))$: et sont telles que $(q_i(\tau'_i), q_{i+1}(\tau_{i+1}))$ est un flanc du graphe, $x_i(\tau'_i)$ appartient à la condition de garde de ce flanc, et $x_{i+1}(\tau_{i+1})$ appartient à l'ensemble de reset de ce flanc.

Dans ce contexte, on peut considérer une condition de garde $G(e)$ comme une validation de la transition $e \in E$: l'exécution peut prendre une transition discrète $e \in E$ à partir d'un état x si $x \in G(e)$. La troisième restriction détermine ce qui arrive durant l'évolution continue, et quand est ce qu'une évolution continue doit donner moyen à une transition discrète.

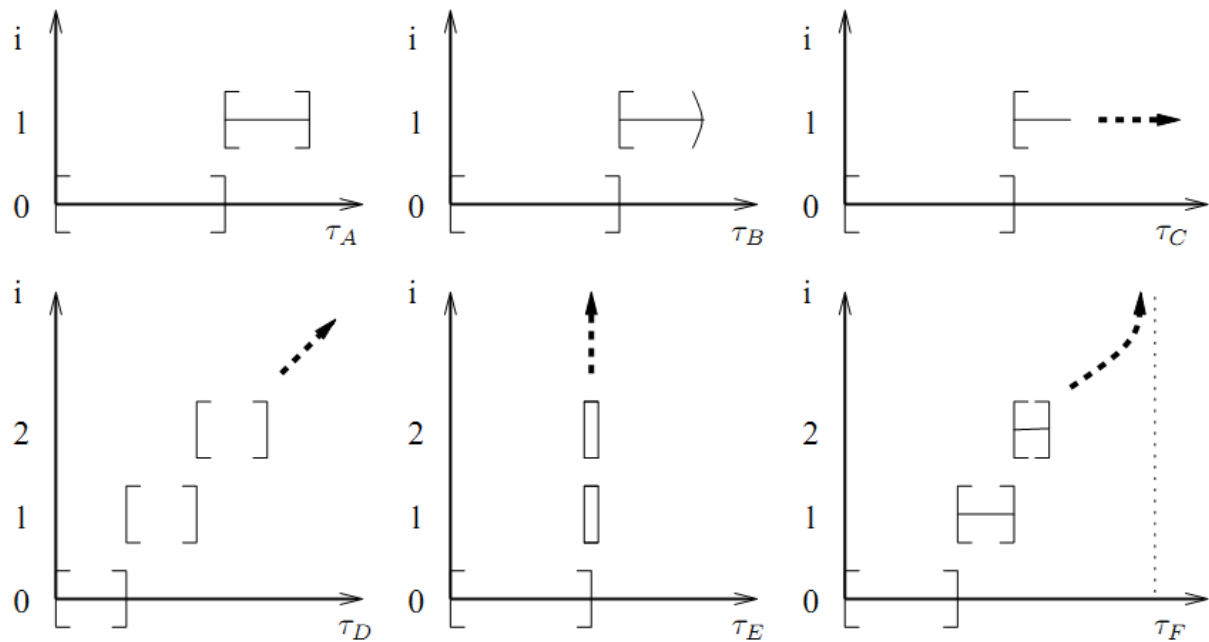
La première partie dicte que pendant l'évolution continue, l'état discret doit rester constant, la deuxième partie stipule que l'état continu doit évoluer selon l'équation différentielle $\dot{x} = f(q, x)$, notons que l'équation différentielle dépend de l'état discret courant (et qui est constant durant l'intervalle d'évolution continue). La troisième partie exige que pendant l'évolution continue, l'état doit rester dans le domaine $\text{Dom}(q)$ de l'état discret. Dans ce contexte, il est commode de considérer $\text{Dom}(q)$ comme forçant les transitions discrètes : l'exécution doit prendre une transition discrète si l'état est sur le point de quitter le domaine. Une interprétation convenable est que les automates hybrides acceptent l'exécution. Cette perspective permet de considérer, par exemple des automates hybrides qui acceptent des exécutions multiples à partir de quelques conditions initiales, une propriété qui peut être utile quand on veut modéliser un système avec des incertitudes.

I.4.2.1 Classification d'exécution hybride[6]

Une exécution (τ, q, x) est appelée :

- **Finie** : si τ est une séquence finie et que le dernier intervalle est fermé.
- **Infinie** : si τ est une séquence infinie, ou si la somme des intervalles de temps dans τ est infini, c.-à-d : $\sum_{i=0}^{\infty} (\tau'_i - \tau_i) = \infty$.
- **Maximale** : si elle n'est pas le préfixe stricte d'aucune autre exécution H .
- **Zeno** : si elle est infinie mais $\sum_{i=0}^{\infty} (\tau'_i - \tau_i) < \infty$

La figure (1.8) montre des exemples d'ensembles de temps hybrides d'exécution finie, infinie et Zeno



Figure(1.8) τ_A finie, τ_C et τ_D infinie, τ_E et τ_F Zeno

I.5 Définition des systèmes dynamiques à commutation [18]

Les systèmes à commutation représentent une classe de systèmes dynamiques hybrides qui fascine par sa simplicité structurelle et par la complexité des phénomènes qu'elle peut décrire. Formellement, un système à commutation en temps continu est défini par la relation :

$$\dot{x}(t) = f_{\sigma(t)}(t, x(t), u(t)) \text{ où}$$

- $\sigma(t): R^+ \rightarrow I = \{1, 2, \dots, N\}$ représente une fonction constante par morceaux, nommée loi de commutation, qui prend des valeurs dans un ensemble d'indices I .
- $x(t) \in R^n$ représente l'état du système,
- $u(t) \in R^m$ la commande, et $f_i(\dots), \forall i \in I$ sont des champs de vecteurs décrivant les différents régimes de fonctionnement du système.

La loi de commutation $\sigma(t)$, spécifie le régime (sous système actif). Seul un sous système est actif à un instant donné. Le choix du sous-système actif peut être lié à un critère temporel, à des régions ou surfaces déterminées dans l'espace d'état, ou à un paramètre extérieur.

I.6 Classes de phénomènes hybrides [7]

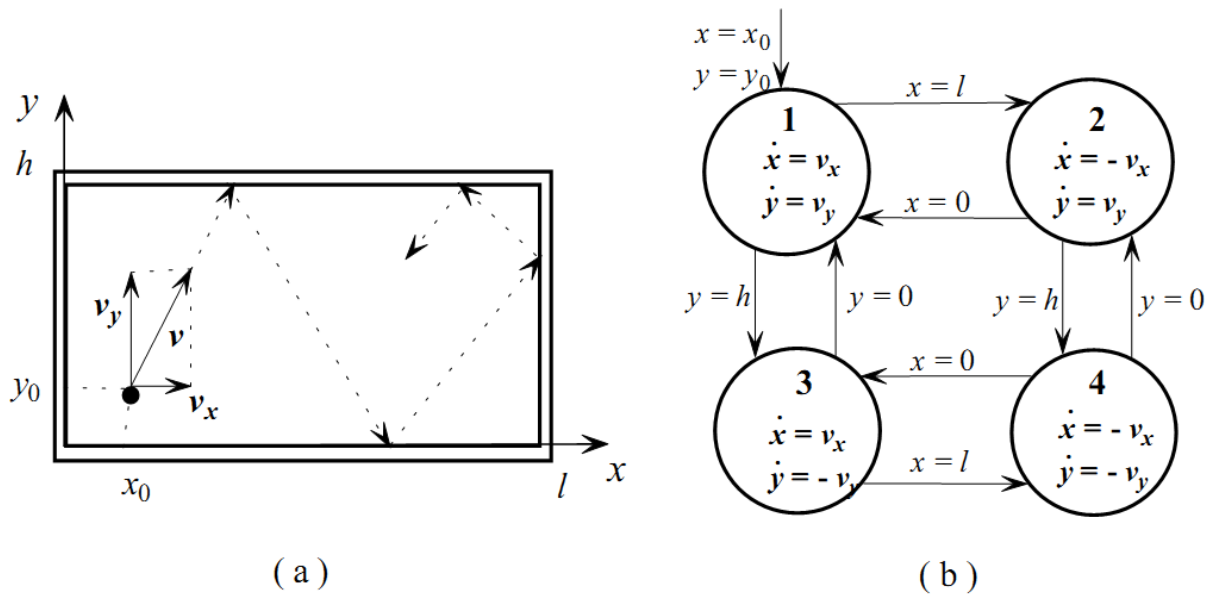
La nature hybride d'un système peut être inhérente aux phénomènes physiques qui le régissent. Un certain nombre de phénomènes physiques considérés comme hybrides ont été regroupés en quatre catégories principales traduisant leur influence sur les modèles mathématiques utilisés pour décrire les différentes classes des systèmes.

I.6.1 Commutation autonome [7]

Une commutation autonome caractérise un phénomène où le champ de vecteur $f(x, t)$ change de façon discontinue lorsque l'état $x(\cdot)$ atteint certains seuils. C'est le cas d'un système à hystérésis.

Le saut autonome est un phénomène similaire rencontré dans les systèmes mécaniques, où l'état $x(\cdot)$ effectue un saut lorsqu'il atteint certaines régions de l'espace, c'est-à-dire qu'il passe de façon discontinue de sa valeur courante à une autre. Un exemple de ce phénomène est donné par la collision de deux corps où la vitesse change brutalement et subit un saut.

Considérons, par exemple, une table de billard de longueur l et de largeur h , avec une boule, comme l'illustre la figure (1.9)(a) [7].



Figure(1.9) (a) trajectoire d'une boule de billard et (b) automate associé

La position initiale de la boule est (x_0, y_0) et après avoir été frappée elle commence à se déplacer avec une vitesse v . Quand la boule arrive à un côté de la table parallèle à l'axe y , elle rebondit et le signe de la composante de la vitesse v_x change.

De même, le signe de la composante de la vitesse v_y change lorsque la boule arrive à un côté parallèle à l'axe x . La combinaison des signes des composantes de la vitesse donne quatre directions différentes du mouvement de la boule.

L'automate modélisant le mouvement de la boule est représenté dans la figure (1.9)(b). Le vecteur de la vitesse peut avoir quatre états différents $[v_x, v_y]^T$, $[-v_x, v_y]^T$, $[v_x, -v_y]^T$ et $[-v_x, -v_y]^T$. Chaque état de la vitesse caractérise une dynamique des variables x et y représentée par un sommet de l'automate.

Le passage d'une dynamique à une autre est modélisée par les arcs de l'automate et se produit lorsque la boule atteint un côté de la table, c'est-à-dire, quand x atteint les valeurs 0 ou l ou y atteint les valeurs 0 ou h .

I.6.2 Commutation contrôlée [7]

Les équations d'évolution changent instantanément en réponse à une commande ou à une modification de l'environnement, donc à une action extérieure au système considéré : c'est le cas lors du passage de l'état de la vitesse 1 à l'état de la vitesse 2.

Le phénomène de commutation contrôlée est illustré à travers l'exemple de l'embrayage mécanique.

Il s'agit d'un système composé de deux masses en rotation. Les masses sont couplées par un embrayage mécanique idéal. Chaque masse i , dont l'inertie est J_i , est entraînée par un couple Q_i à une vitesse de rotation ω_i . Quand les masses sont couplées, les valeurs des vitesses de rotation sont identiques. Ces vitesses sont indépendantes quand les masses sont découplées (Figure 1.10) [7].

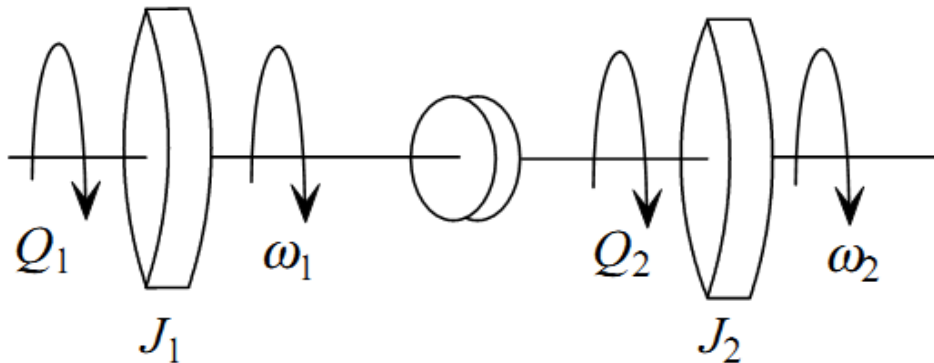


Figure (1.10) Système d'embrayage mécanique

Lorsque les axes de rotation sont indépendants le système est débrayé. Le modèle mathématique est donné par :

$$\begin{pmatrix} J_1 & 0 \\ 0 & J_2 \end{pmatrix} \begin{pmatrix} \dot{\omega}_1 \\ \dot{\omega}_2 \end{pmatrix} = \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} \omega_1 \\ \omega_2 \end{pmatrix} + \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} Q_1 \\ Q_2 \end{pmatrix}$$

Quand les axes de rotation sont couplés le système est embrayé, les vitesses de rotation sont identiques et le système peut être décrit par l'équation :

$$\begin{pmatrix} J_1 & J_2 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} \dot{\omega}_1 \\ \dot{\omega}_2 \end{pmatrix} = \begin{pmatrix} 0 & 0 \\ 1 & -1 \end{pmatrix} \begin{pmatrix} \omega_1 \\ \omega_2 \end{pmatrix} + \begin{pmatrix} 1 & 1 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} Q_1 \\ Q_2 \end{pmatrix}$$

D'une manière plus générale, le système peut donc être décrit par une équation sous la forme:

$$I_k \dot{x} = A_k x + B_k u$$

Où les matrices I_k , A_k et B_k dépendent de l'état k du système. Nous avons donc une équation d'état dans laquelle figure la représentation hybride d'écrivant la dynamique continue du système dans ses différents états.

I.6.3 Saut autonome [2]

Cette fois-ci l'état ou certaines de ses composantes changent brusquement de valeur lorsqu'une valeur de référence est atteinte. Ainsi, la vitesse d'un mobile s'annule instantanément, voire change de signe, en cas de choc.

Il s'agit là également d'une approximation théorique en considérant les différentes échelles de temps impliquées, car si cette présentation du phénomène physique traduisait parfaitement la réalité, il y aurait une variation d'énergie infinie au moment du saut.

I.6.4 Saut contrôlé [2]

Cette fois l'état change instantanément sous l'influence d'une action extérieure. Ainsi, le montant d'un compte rémunéré accuse-t-il une évolution progressive et des sauts contrôlés en cas de versement ou de retrait. Il faut là encore remarquer que pour des systèmes physiques, le changement s'opère à une autre échelle de temps que celle du phénomène étudié.

Conclusion :

Les systèmes dynamiques hybrides ont une grande importance dans plusieurs secteurs industriels, ainsi que dans beaucoup d'autres activités humaines. Ils comportent une partie à évolution continue dans le temps modélisée par des équations différentielles et une autre à évolution événementielle modélisée par un automate fini.

Dans ce chapitre nous avons élaboré les définitions de bases des fonctionnements des systèmes dynamiques hybrides. Nous avons présenté le modèle hybride et expliqué en détail son fonctionnement. On a donné le modèle de thermostat comme exemple pour expliquer le fonctionnement d'un automate hybride. Le choix de ce dernier s'est avéré complètement efficace pour la modélisation de la complexité structurelle des systèmes dynamiques hybrides.

Introduction [8]

De nombreux systèmes physiques, chimiques et économiques, peuvent être modélisés par des relations mathématiques. Ces systèmes peuvent changer avec le temps ou toute autre variable indépendante selon des relations dynamiques. Il est possible d'orienter ces systèmes d'un état à un autre par l'application d'un certain type d'entrées externes ou des contrôles. Si tout cela peut être fait, il peut y avoir des façons différentes de faire la même tâche. S'il y a des manières différentes de faire la même tâche, donc il peut y avoir une meilleure façon de la faire. Cette meilleure façon peut être d'aller d'un état à un autre en un temps minimum, ou bien par exemple développée une poussée maximale d'un moteur d'une fusée.

L'entrée donnée au système correspondant est appelée «contrôle optimal». La mesure de l'exécution est appelée «indice de performance» ou «fonction de coût». Ainsi, on obtient un système de contrôle optimal, quand un système est commandé de manière optimale atteignant des objectifs fixés. L'intérêt théorique et les aspects pratiques de la commande optimale ont été vite appliqués dans des domaines aussi variés que l'énergie électrique, l'aérospatiale, les usines chimiques, l'économie, la biologie et l'écologie.

II.1 Définition du Problème de commande optimale [9]

Considérons un système de contrôle général :

$$\dot{x}(t) = f(t, x(t), u(t)) \quad , x(t_0) = x_0 \quad (2,1)$$

Où f est une application de classe C^1 de $I \times V \times U$ dans R^n , I est un intervalle de R , V est un sous ensemble ouvert de R^n et U est un sous ensemble ouvert de R^m , $(t_0, x_0) \in I \times V$. Par ailleurs on suppose que les contrôles $u(\cdot)$ appartiennent à un sous ensemble de $L_{loc}^\infty(I, R^m)$.

on suppose aussi que $t_0 = 0$.

Pour tout contrôle $u \in L_{loc}^\infty(I, R^m)$, la trajectoire associée $x_u(t)$ est définie sur un intervalle maximal $[0, t_e(u)]$, où $t_e(u) \in R^+ \cup \{+\infty\}$.

Chapitre II

Commande optimale des systèmes dynamiques

Pour tout $T > 0, T \in I$ on note U_T l'ensemble des contrôles admissibles sur $[0, T]$, c'est-à-dire l'ensemble des contrôles tels que la trajectoire associée soit bien définie sur $[0, T]$, autrement dit $T < T_e(u)$.

Soient f une fonction de classe C^1 sur $I \times V \times U$, et g une fonction continue sur V . Pour tout contrôle $u \in U_T$ on définit le coût de la trajectoire associée à $x_u(\cdot)$ sur l'intervalle $[0, T]$

$$C(T, u) = \int_0^T f(t, x_u(t), u(t)) dt + g(T, x_u T) \quad (2,3)$$

Soient M_0 et M_1 deux sous-ensembles de V . Le problème de contrôle optimal est de déterminer les trajectoires $x_u(\cdot)$ solutions de :

$$\dot{x}(t) = f(t, x(t), u(t))$$

Telles que $x_u(0) \in M_0, x_u(T) \in M_1$, et minimisant le coût $C(T, u)$. On dit que le problème de contrôle optimal est à temps final non fixé si le temps final T est libre, sinon on parle de problème à temps final fixé.

II.2 Existence de trajectoires optimales [9]

Pour déterminer une trajectoire optimale joignant un ensemble initial à une cible, il faut d'abord savoir si cette cible est atteignable. C'est le problème de contrôlabilité. Ensuite, une fois ce problème résolu, il faut chercher parmi toutes ces trajectoires possibles celles qui le font en un coût minimum.

II.2.1 Contrôlabilité

II.2.1.1 Contrôlabilité des systèmes généraux [9]

On veut répondre à la question suivante : étant donné le système (2.1), où peut-on aller en un temps T en faisant varier le contrôle u ?

Définition II.1 L'ensemble accessible en temps T pour le système (2.1), noté $\text{Acc}(x, T)$, est l'ensemble des extrémités au temps T des solutions du système partant de x_0 au temps $t = 0$.

Chapitre II

Commande optimale des systèmes dynamiques

Théorème II.1 Considérons le système de contrôle :

$$\dot{x}(t) = f(t, x(t), u(t)) \quad , x(t_0) = x_0$$

où la fonction f est de C^1 sur R^{1+n+m} , et les contrôles u appartiennent à l'ensemble U des fonctions mesurables à valeurs dans un ensemble compact $\Omega \subset R^m$ (Ω est un intervalle contenant l'origine). On suppose que :

- il existe un réel positif b tel que toute trajectoire associée est uniformément bornée par b sur $[0, T]$:

$$\exists b > 0 \mid \forall u \in U, \forall t \in [0, T], \|x_u(t)\| \leq b, \quad (2,2)$$

-pour tout (t, x) , l'ensemble des vecteurs vitesses

$$V(t, x) = \{f(t, x, u) \mid u \in \Omega\} \quad (2,3)$$

est convexe.

Alors l'ensemble $Acc(x_0, t)$ est compact et varie continûment en t sur $[0, T]$.

Définition II.2 Le système (2.1) est dit contrôlable (en temps quelconque) depuis x_0 si :

$$R^n = \bigcup_{T \geq 0} Acc(x_0, T)$$

Il est dit contrôlable en temps T si $R^n = Acc(x_0, T)$.

II.2.1.2 Contrôlabilité des systèmes affines [10]

Proposition II.1 Considérons le système affine dans R^n

$$\dot{x} = f_0(x) + \sum_{i=1}^m u_i f_i(x), \quad x(0) = x(T) = x_1 \quad (2,4)$$

Avec le coût

$$J_t(u) = \int_0^T \sum_{i=1}^m u_i^2(t) dt \quad (2,5)$$

Où $T > 0$ est fixé et la classe U des contrôles admissibles est le sous-ensemble de $L^2([0, T], R^m)$ tel que

1. $\forall u \in U$ x_u est bien définie sur $[0, T]$.
2. $\exists B_T \mid \forall u \in U \quad \forall t \in [0, T] \quad \|x_u(t)\| \leq B_T$.
3. Si x_1 est accessible depuis x_0 en temps T , alors il existe un contrôle optimal u reliant x_0 et x_1 .

II.3 Les deux grands principes de la commande optimale [14]

Deux grands principes permettent de traiter les problèmes de commande optimale : le principe du maximum de Pontryaguine et le principe d’optimalité de Bellmann. Ce dernier s’applique sur tout système dynamique muni d’un critère additif et conduit à la programmation dynamique sur des équations de récurrence. On l’utilise principalement sur des problèmes formulés en temps discret.

Les méthodes variationnelles s’appliquent lorsqu’il est possible d’évaluer la variation du critère en fonction de la variation de la commande. L’idée de base est de caractériser la commande optimale u^* par le fait qu’une commande $u = u^* + \delta u$ doit donner un critère supérieur $J = J^* + \delta J > J^*$. En exprimant δJ en fonction de δu , on peut espérer trouver des caractérisations de u^* .

II.4 Principe du maximum de Pontriaguine [9]

II.4.1 Enoncé général

Théorème II.2 Considérons le système de contrôle dans R^n

$$\dot{x}(t) = f(t, x(t), u(t)) \tag{2.6}$$

Où $f: R \times R^n \times R^m \rightarrow R^n$ est de classe C^1 , U l’ensemble des contrôles admissibles u dont les trajectoires associées relient un point initial x_0 à un point final $0 \in R^n$.

On note U l’ensemble des contrôles admissibles u dont les trajectoires associées relient un point initial de M_0 à un point final M_1 en temps $t(u) < t_e(u)$.

Par ailleurs on définit le coût d’un contrôle u sur $[0, t]$

$$J(t, u) = \int_0^t f^0(s, x(s), u(s)) ds + g(t, x(t))$$

Où $f^0: R \times R^n \times R^m \rightarrow R$ et $g: R \times R^n \rightarrow R$ sont de C^1 , et $x(\cdot)$ est la trajectoire solution de (2.6) associé au contrôle u .

Si le contrôle $u \in U$ associé à la trajectoire $x(\cdot)$ est optimal sur $[0, T]$, alors il existe une application $p(\cdot): [0, T] \rightarrow R^n$ absolument continue appelée vecteur adjoint, et un réel $p^0 \leq 0$ tels que le couple $(p(\cdot), p^0)$ est non trivial, et tels que, pour presque tout $t \in [0, T]$,

$$\begin{cases} \dot{x}(t) = \frac{\partial H}{\partial p}(t, x(t), p(t), p^0, u(t)) \\ \dot{p}(t) = \frac{\partial H}{\partial x}(t, x(t), p(t), p^0, u(t)) \end{cases} \quad (2.7)$$

$$\text{Où } H(t, x, p, p^0, u) = \{p, f(t, x, u)\} + p^0 f^0(t, x; u)$$

Est l'Hamiltonien du système, et on a la condition de maximisation presque partout sur $[0, T]$.

$$H(t, x(t), p(t), p^0, u(t)) = \max_{v \in \Omega} H(t, x(t), p(t), p^0, v) \quad (2.8)$$

Si de plus le temps final pour joindre la cible M_1 n'est pas fixé, on a la condition au temps final T .

$$\max_{v \in \Omega} H(T, x(T), p(T), p^0, v) = -p^0 \frac{\partial g}{\partial t}(T, x(T)) \quad (2.9)$$

Si de plus M_0 et M_1 (ou juste l'un des deux ensembles) sont des variétés de R^n ayant des espaces tangents en $x(0) \in M_0$ et $x(T) \in M_1$, alors le vecteur adjoint peut être construit de manière à vérifier les conditions de transversalité aux deux extrémités (ou juste l'une des deux)

$$p(0) \perp T_{x(0)}M_0 \quad (2.10)$$

Et

$$p(T) - p^0 \frac{\partial g}{\partial t}(T, x(T)) \perp T_{x(T)}M_1 \quad (2.11)$$

Remarque II.1 si le contrôle u est continu au temps T , la condition (2.9) peut s'écrire

$$H(T, x(T), p(T), p^0, u(T)) = -p^0 \frac{\partial g}{\partial t}(T, x(T)) \quad (2.12)$$

Remarque II.2 si la variété M_1 s'écrit sous la forme :

$$M_1 = \{x \in R^n \mid F_1(x) = \dots = F_p(x) = 0\}$$

Où les F_i sont des fonctions de classe C^1 sur R^n (indépendantes puisque M_1 est une variété), alors la condition (2.11) se met sous la forme :

$$\exists \lambda_1, \dots, \lambda_p \in R \mid p(T) = \sum_{i=1}^p \lambda_i \nabla F_i(x(T)) + p^0 \frac{\partial g}{\partial x}(T, x(T)) \quad (2.13)$$

Remarque II.3 Dans les conditions du théorème, on a de plus pour presque tout $t \in [0, T]$:

$$\frac{d}{dt} H(t, x(t), p(t), p^0, u(t)) = \frac{dH}{dt}(t, x(t), p(t), p^0, u(t)) \quad (2.14)$$

En particulier si le système augmenté est autonome, si f et f^0 ne dépendent pas de t , alors H ne dépend pas de t , et on a :

$$\forall t \in [0, T] \quad \max_{v \in \Omega} H(x(t), p(t), p^0, v) = cste$$

Chapitre II

Commande optimale des systèmes dynamiques

Notons que cette égalité est alors valable partout sur $[0, T]$ (en effet cette fonction de t est lipschitzienne)

Remarque II.4 La convention $p^0 \leq 0$ conduit au principe du maximum. La convention $p^0 \geq 0$ conduirait au principe du minimum, la condition (2.8) serait une condition de minimum.

Remarque II.5 dans le cas où $\Omega = R^m$, lorsqu'il n'y a pas de contraintes sur le contrôle, la condition de maximum (2.8) devient $\frac{dH}{du} = 0$.

Définition II.3 Une extrémale du problème de contrôle optimal est un quadruplet $(x(\cdot), p(\cdot), p^0, u(\cdot))$ solution des équations (2.7) et (2.8). Si $p_0 = 0$, on dit que l'extrémale est anormale, et si $p^0 \neq 0$ l'extrémale est dite normale.

Définition II.4 Les conditions (2.10) et (2.11) sont appelées conditions de transversalité sur le vecteur adjoint. La condition (2.9) est appelée condition de transversalité sur l'Hamiltonien.

Remarque II.6 Le problème important du temps minimal correspond à

$f^0 = 1$ et $g = 0$, ou bien à $f^0 = 0$, et $g(t, x) = t$. Dans les deux cas les conditions de transversalité obtenues sont bien les mêmes.

II.5 Programmation dynamique [8]

Compte tenu d'un processus dynamique, il y a essentiellement deux façons de résoudre un problème de contrôle optimal, l'un est le principe du maximum de Pontriaguine (vu précédemment), et l'autre est la programmation dynamique de Bellman. Ce dernier conduit à la programmation dynamique. La technique est appelée programmation dynamique, car il s'agit d'une technique basée sur l'ordinateur (de programmation) et adaptée à des systèmes dynamiques. L'idée de base de la programmation dynamique est un problème d'optimisation discret à plusieurs étapes dans le sens où, à chaque ensemble fini, une décision est choisie parmi un ensemble fini d'un certain nombre de décisions basées sur un critère d'optimisation. Le thème central de la programmation dynamique est basé sur un concept simple et intuitif appelée principe d'optimalité.

II.5.1 Principe d'optimalité [8]

Considérons un simple processus de décision à plusieurs étapes d'optimisation montré dans la Figure (2.1)

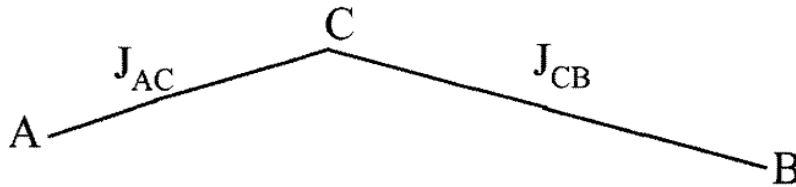


Figure (2.1) Chemin optimal de A à B

Le coût d'optimisation pour l'ensemble du segment AB est donné par l'équation :

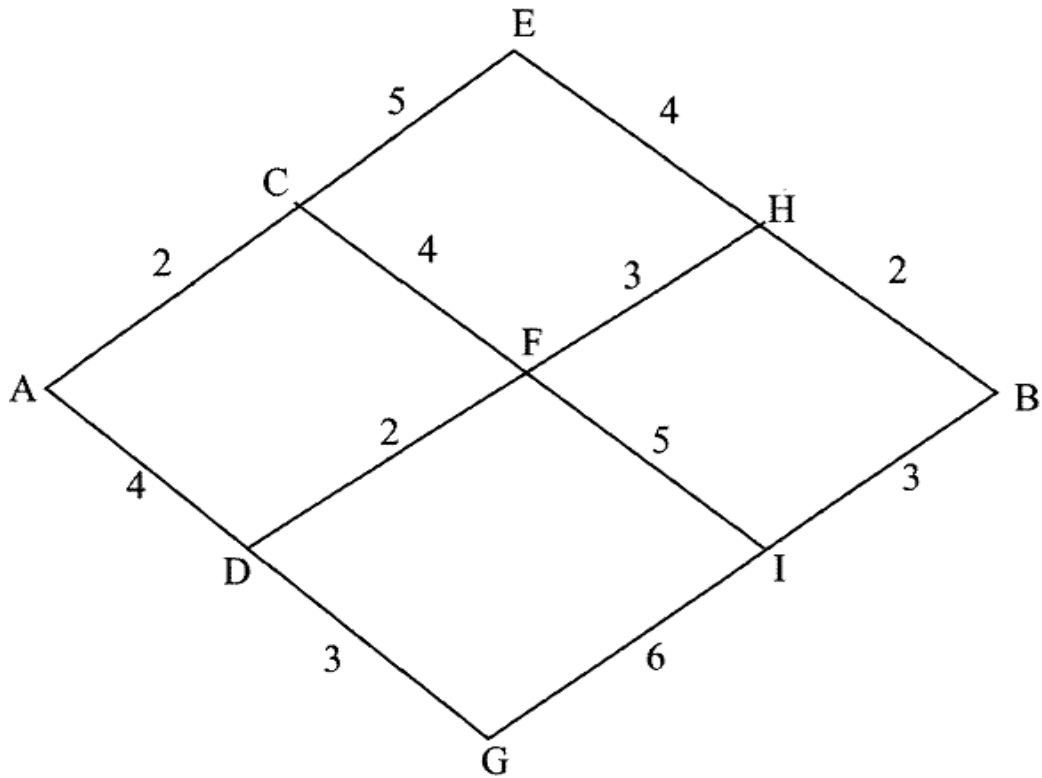
$$J_{AB} = J_{AC} + J_{CB} \quad (2.15)$$

Si J_{AC} est le coût optimal du segment AC , alors J_{CB} est le coût optimal du segment restant CB . En d'autres termes, on peut décomposer la totalité du chemin optimal totale en segments plus petits qui sont eux-mêmes optimaux. Inversement, si on trouve des valeurs optimales pour ces segments plus petits, alors on peut obtenir la valeur optimale pour l'ensemble du chemin. Cette propriété évidente à la recherche est appelée principe d'optimalité et déclarée comme suit :

Une politique optimale a la propriété que quelque soit l'état et la décision (la commande) initiaux, les décisions restantes doivent constituer une politique optimale en ce qui concerne l'état résultant de la décision précédente.

II.5.1.1 Solution arrière [8]

Il semble naturel de commencer à travailler de l'arrière, bien que l'on puisse également avancer les travaux de la phase initiale. Pour illustrer le principe d'optimalité, nous considérons un processus de décision en plusieurs étapes comme le montre la figure (2.2).



Figure(2.2) processus de décision à plusieurs étapes

La figure (2.2) Représente un processus de décision à plusieurs étapes. Dans un système de routage d'un avion, les deux points A,B représentent respectivement les deux villes de départ et d'arrivée, les nœuds C, D, E, F, G, H et I représentent les villes intermédiaires entre les deux points A et B. Les numéros (appelés unités) sur chaque segment indiquent le coût (ou l'indice de performance) de l'avion entre les deux villes.

Maintenant, nous sommes intéressés à trouver l'itinéraire le plus économique de voler de la ville A vers la ville B. Nous avons 5 étapes à partir de $k = 0$ à $k = N = 4$. En outre, on peut associer l'état actuel à la jonction ou le nœud.

La décision est prise à chaque état, la décision ou le contrôle est $u = \pm 1$, où $u = + 1$ indique le déplacement vers le haut ou vers la gauche et $u = -1$ indique le déplacement vers le bas ou vers la droite en regardant de chaque jonction vers la droite. Le Travail de l'algorithme de programmation dynamique est montré dans la figure (2.3).

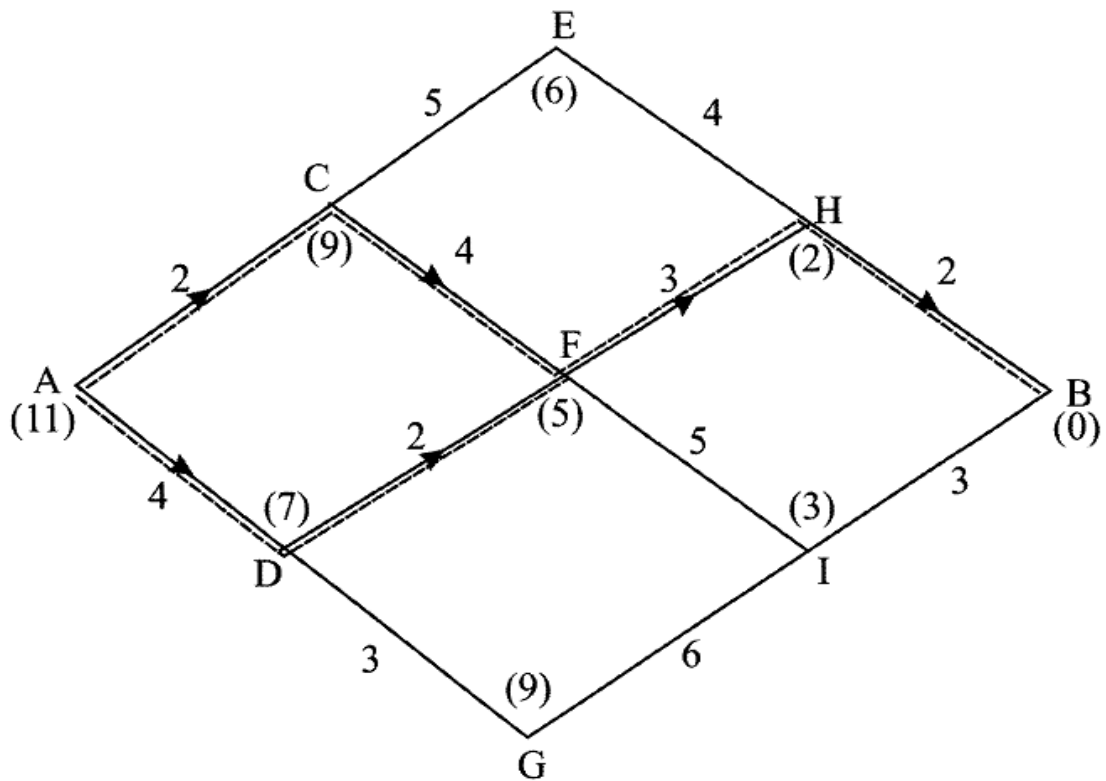


Figure (2.3) Processus de décision à plusieurs étapes: Solution arrière

Etape 5 : $K = K_f = N = 4$

Ce n'est que le point de départ, il n'y a qu'une seule ville B et par conséquent il n'y a pas de coût.

Etape 4 : $k=3$

Il y a deux villes H et I, à ce stade, nous devons trouver l'itinéraire le plus économique de cette étape à l'étape 5. On travaille en arrière, nous commençons avec B qui peut être atteinte par H ou I. L'avion prend 2 unités à voler de H à B en utilisant le contrôle ou la décision $u = -1$ (ou à la baisse droite) et donc nous plaçons le numéro 2 entre parenthèses sous H. De même, il faut 3 unités de voler de I à B en utilisant le contrôle ou décision $u = +1$ (vers le haut ou vers la gauche) et donc de placer le numéro 3 vient de proximité de I. Laissez-nous également placer une tête de flèche aux chemins correspondants ou les routes. Notez qu'il n'y a pas d'autre moyen de voler à partir de H vers B et I vers B, sauf comme indiqué par les flèches.

Chapitre II

Commande optimale des systèmes dynamiques

Etape 3 : $k=2$

Ici, il ya trois villes E, F, G et de ces nœuds, nous pouvons voler à H et I. Considérons d'abord E, le coût total de voler de E à B sera de $2 + 4 = 6$ en utilisant le contrôle ou la décision $u = -1$ (ou à la baisse droite) et laissez-nous placer 6 unités entre parenthèses à la ville E. Deuxièmement, à partir de F, nous pouvons prendre deux voies F, H, B et F, I, B, en utilisant des décisions $u = +1$ (à la hausse ou à gauche) et $u = -1$ (vers le bas ou à droite) et les coûts correspondants sont $2 + 3 = 5$ et $3 + 5 = 8$, respectivement. Remarquez que nous avons placé 5 au lieu de 8 au nœud F et une tête de flèche sur le segment pour indiquer le coût optimal de voler par F, H, B au lieu de F, I, B. Enfin, si nous pensons au nœud G, il n'ya qu'une seule route qui est G, I, B pour aller à B à partir de G. on constate que le nœud G possède le coût maximum 9.

Etape 2 : $k = 1$

Par la même procédure comme expliqué dans les cas précédents, nous voyons que le nœud C a un coût minimum 9 et le nœud D a un coût minimum 7.

Etape 1 : $k=0$

Ici, notez que de A, les deux segments AC et AD ont le même coût minimum.

Solution optimale

C'est facile à trouver, nous venons de suivre la route des têtes de flèches à partir de A vers B. Notez qu'il existe deux routes optimale de joindre les deux villes. Ainsi, la route la plus économique est soit A, C, F, H, B ou A, D, F, H, B. Le coût total minimum est de 11 unités.

II.5.1.2 Solution avant [8]

On peut résoudre le système précédent en utilisant une solution de l'avant, nous obtiendrons un résultat identique à la solution en arrière comme le montre la figure (2.4).

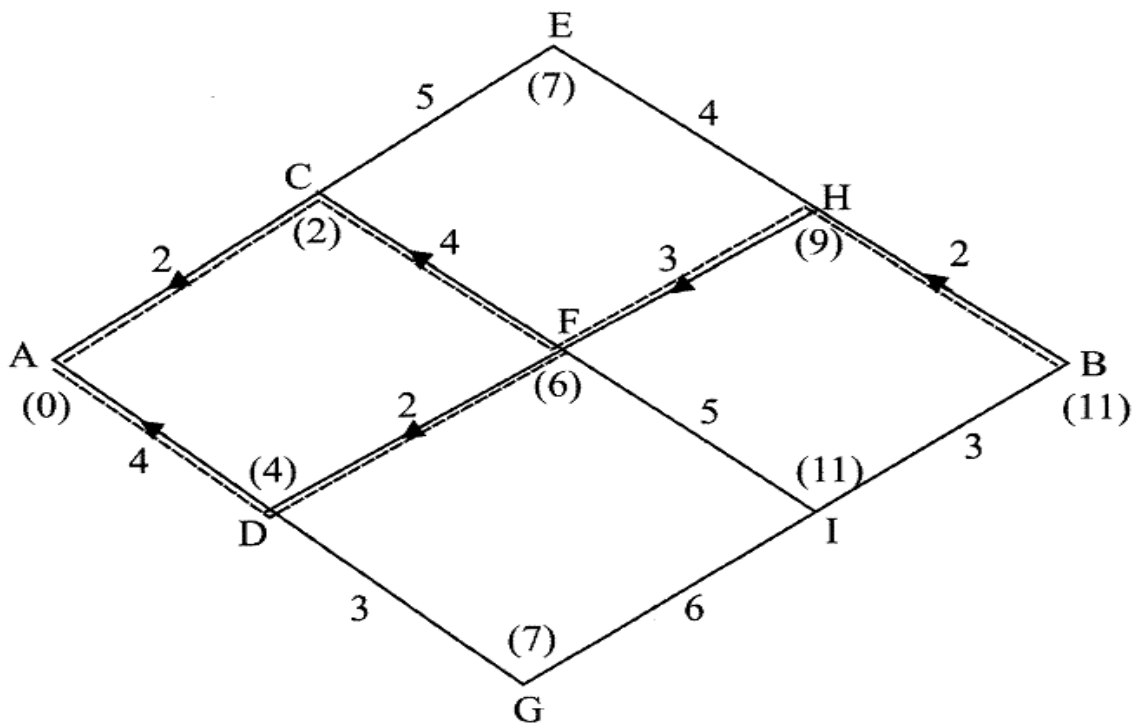


Figure (2.4) Processus de décision à plusieurs étapes

Ainsi, comme illustré dans les deux cas précédents, nous avons :

- 1. diviser la totalité du parcours en plusieurs étapes.
- 2. Trouver l'itinéraire optimal (économique) de chaque étape.
- Enfin, en utilisant le principe d'optimalité, nous sommes en mesure de combiner les différents segments optimaux dans un seul itinéraire optimal unique (ou trajectoire).

Dans les cas précédents, nous avons fixé à la fois les points initiaux et finaux et donc nous avons un système de point final fixe. Nous pouvons de même s'adresser au point extrême du système.

Ensuite, nous explorons la façon dont le principe d'optimalité de la programmation dynamique peut être utilisé pour les systèmes de contrôle optimal. Nous remarquons que l'approche de la programmation dynamique est naturellement un système à temps discret. En outre, elle peut être facilement appliquée soit à des systèmes linéaires ou non linéaires.

II.5.2 Contrôle optimal utilisant la programmation dynamique [8]

Considérons d'abord le contrôle optimal d'un système à temps discret, ceci ne constitue pas une perte de généralité car même si le système est à temps continu, on peut facilement le discrétiser en utilisant l'une des plusieurs approches. Le système est décrit par l'équation suivante :

Chapitre II

Commande optimale des systèmes dynamiques

$$x(k + 1) = f(x(k), u(k), k) \quad (2.15)$$

Et soit la fonction coût

$$J_i(x(k_i)) = J = S(x(k_f), k_f) + \sum_{k=i}^{k_f-1} V(x(k), u(k)) \quad (2.16)$$

Où, $x(k), u(k)$ sont les états et les vecteurs de commande respectivement.

Nous sommes intéressés à utiliser le principe d'optimalité pour trouver le contrôle optimal $u^*(k)$ qui s'applique à l'équation (2.16) et donne l'état optimal $x^*(k)$.

Supposons que nous avons évalué le contrôle optimal, l'état et le coût pour toutes les valeurs à partir de $k + 1$ à k_f . Puis, à tout moment de l'étape k , nous utilisons le principe d'optimalité d'écrit par

$$J_k^*(x(k)) = \min_{u(k)} [V[x(k), u(k)] + J_{k+1}^*(x^*(k + 1))] \quad (2.17)$$

La relation précédente est la forme mathématique du principe d'optimalité appliquée au système de commande optimale. Elle est aussi appelé équation fonctionnelle de la programmation dynamique.

Ainsi, cela signifie que si on avait trouvé la commande optimale, l'état et le coût de tout étape $k + 1$ à la phase finale k_f , alors on peut trouver les valeurs optimales pour une seule étape à partir de k à $k + 1$.

Exemple [8]

Prenons un simple exemple d'un scalaire pour illustrer la procédure qui explique la méthode de la programmation dynamique :

$$x(k + 1) = x(k) + u(k) \quad (2.18)$$

Le critère de performance à optimiser est

$$J = \frac{1}{2} x^2(k_f) + \frac{1}{2} \sum_{k=k_0}^{k_f-1} [x^2(k) + u^2(k)] \quad (2.19)$$

Chapitre II

Commande optimale des systèmes dynamiques

Pour la simplicité des calculs, nous prendrons $k_f = 2$, les contraintes et les valeurs de quantification sur le contrôle sont d'écrite par :

$$\begin{cases} -1 \leq u(k) \leq +1, & k = 0,1,2 \text{ où} \\ u(k) = -1, 0.5, & 0, 0.5, 1 \end{cases} \quad (2.20)$$

Et sur l'état

$$\begin{cases} 0 \leq x(k) \leq +2, & k = 0, 1 \text{ où} \\ x(k) = 0, 0.5, & 1, 1.5, 2 \end{cases} \quad (2.21)$$

Trouver la séquence de contrôle optimal $u^*(k)$ et l'état $x^*(k)$ qui minimisent le critère de performance (2.19).

Solution: Pour utiliser le principe d'optimalité pour résoudre le système précédent, nous avons d'abord mis en place une grille entre $x(k)$ et k , nous divisons les étapes en deux ensembles : l'un pour $k = 2$ et l'autre pour $k = 1$.

Nous commençons avec $k = 2$ et on va d'abord trouver les valeurs optimales et travailler à rebours pour $k = 1$, en utilisant l'état (2.18), la fonction de coût (2.19) et le contrôle optimal (2.20).

Etape: $k = 2$

Calculons d'abord l'état $x(2)$ en utilisant la relation d'état (2.18) pour toutes les valeurs admissibles de $x(k)$ et $u(k)$ données par (2.20) et (2.21). Ainsi, par exemple, pour la valeur admissible de $x(1) = 2$ et $u(1) = -1, -0.5, 0.5, 1$, nous avons :

$$\left\{ \begin{array}{l} x(2) = x(1) + u(1) \\ x(2) = 2 + (-1) = 1 \\ x(2) = 2 + (-0.5) = 1.5 \\ x(2) = 2 + 0 = 2 \\ x(2) = 2 + 0.5 = 2.5 \\ x(2) = 2 + 1 = 3 \end{array} \right. \quad (2.22)$$

Notez les valeurs 2.5 et 3 (représentées en rouge) de l'état $x(2)$ ne sont pas autorisés en raison du dépassement de la contrainte étatique (2.21).

En outre, de l'équation fonctionnelle (2.17), nous avons pour cet exemple :

$$J_k(x(k)) = \min_{u(k)} \left[\frac{1}{2} u^2(k) + \frac{1}{2} x^2(k) + J_{k+1}^* \right] \quad (2.23)$$

À partir de laquelle nous avons pour le coût optimal à $k = 2$

$$J_{k_f}^* = \frac{1}{2}x^2(2) \tag{2.24}$$

Qui est évalué pour toutes les valeurs admissibles de $x(2)$ que

$$\begin{aligned} J_{k_f}^* &= 2 && \text{pour } x(2) = 2 \\ &= 1.125 && \text{pour } x(2) = 1.5 \\ &= 0.5 && \text{pour } x(2) = 1 \\ &= 0.125 && \text{pour } x(2) = 0.5 \\ &= 0 && \text{pour } x(2) = 0 \end{aligned} \tag{2.25}$$

Les calculs entiers sont présentés dans le tableau (2.1) pour $k = 2$ et en Tableau (2.2) pour $k = 1$. Les données des tableaux (2.1) et (2.2) correspondant à des conditions optimales sont représentées dans le cadre de la programmation dynamique dans la Figure (2.5) Ici, dans cette figure, $u_0^* = u^*(x(0), 0)$, et $u_1^* = u^*(x(1), 1)$ et les quantités entre parenthèses sont les valeurs de coût optimales. Par exemple, à l'étape $k = 1$ et l'état $x(k) = 1$, la valeur $J_{12}^* = 0.75$ indique que le coût de transfert de l'état $x(1)$ vers $x(2)$.

Ainsi, dans la figure (2.5), pour trouver les trajectoires optimales pour tout état initial, nous avons simplement suivi les flèches. Par exemple, pour transférer l'état $x(0) = 1$ à l'état $x(2) = 0$, nous devons en premier appliquer $u_0^* = -1$ pour le transférer à $x(1) = 0$ puis $u_1^* = 0$.

Tableau (2.1) Calcul du coût au cours de la dernière étape $k = 2$

état actuel $x(1)$	Control actuel $u(1)$	Prochain état $X(2)$	Coût J_{12}	Coût optimal $J_{12}^*(x(1))$	Control optimal $u^*(x(1), 1)$
2	-1	1	3	$J_{12}^*(2) = 2.25$	$u^*(2, 1) = -0.5$
	-0.5	1.5	2.25		
	0	2	4		
	0.5	2.5			
	1	3			
1.5	-1	0.5	1.75	$J_{12}^*(1.5) = 1.75$ $J_{12}^*(1.5) = 1.75$	$u^*(1.5, 1) = -1$ $u^*(1.5, 1) = -0.5$
	-0.5	1	1.75		
	0	1.5	2.25		
	0.5	2	3.25		
	1	2.5			
1	-1	0	1	$J_{12}^*(1) = 0.75$	$u^*(1, 1) = -0.5$
	-0.5	0.5	0.75		
	0	1	1		
	0.5	1.5	1.75		
	1	2	3		
0.5	-1	-0.5		$J_{12}^*(0.5) = 0.25$ $J_{12}^*(0.5) = 0.25$	$u^*(0.5, 1) = -0.5$ $u^*(0.5, 1) = 0$
	-0.5	0	0.25		
	0	0.5	0.25		
	0.5	1	0.75		
	1	1.5	1.75		
0	-1	-1		$J_{12}^*(0) = 0$	$u^*(0, 1) = 0$
	-0.5	-0.5			
	0	0	0		
	0.5	0.5	0.25		
	1	1	1		

Ce tableau est utilisé pour calculer $x(0)$ avec : $x(2) = x(1) + u(1)$;

$$J_{12} = 0.5x^2(2) + 0.5u^2(1) + 0.5x^2(1).$$

La couleur rouge indique que la valeur n'est pas recevable.

Tableau (2.2) Calcul du coût au cours de la dernière étape k = 1

état actuel $x(0)$	Control actuel $u(0)$	Prochain état $X(1)$	Coût J_{02}	Coût optimal $J_{02}^*(x(0))$	Control optimal $u^*(x(0), 0)$
2	-1	1	3.25	$J_{02}^*(2) = 3.25$	$u^*(2,0) = -1$
	-0.5	1.5	3.875		
	0	2	4.25		
	0.5	2.5			
	1	3			
1.5	-1	0.5	1.875	$J_{02}^*(1.5) = 1.875$	$u^*(1.5,0) = -1$
	-0.5	1	2		
	0	1.5	2.875		
	0.5	2	3.25		
	1	2.5			
1	-1	0	1	$J_{02}^*(1) = 1$	$u^*(1,0) = -1$
	-0.5	0.5	0.875		
	0	1	1.25		
	0.5	1.5	2.375		
	1	2	3		
0.5	-1	-0.5		$J_{02}^*(0.5) = 0.25$	$u^*(0.5,0) = -0.5$
	-0.5	0	0.25		
	0	0.5	0.375		
	0.5	1	1		
	1	1.5	2.375		
0	-1	-1		$J_{02}^*(0) = 0$	$u^*(0,0) = 0$
	-0.5	-0.5			
	0	0	0		
	0.5	0.5	0.375		
	1	1	1.25		

Utilisez-les pour calculer le ci-dessus: $x(1) = x(0) + u(0)$;

$$J_{02} = 0.5u^2(0) + 0.5x^2(0) + J_{12}^*(x(1))$$

La couleur rouge indique que la valeur n'est pas recevable.

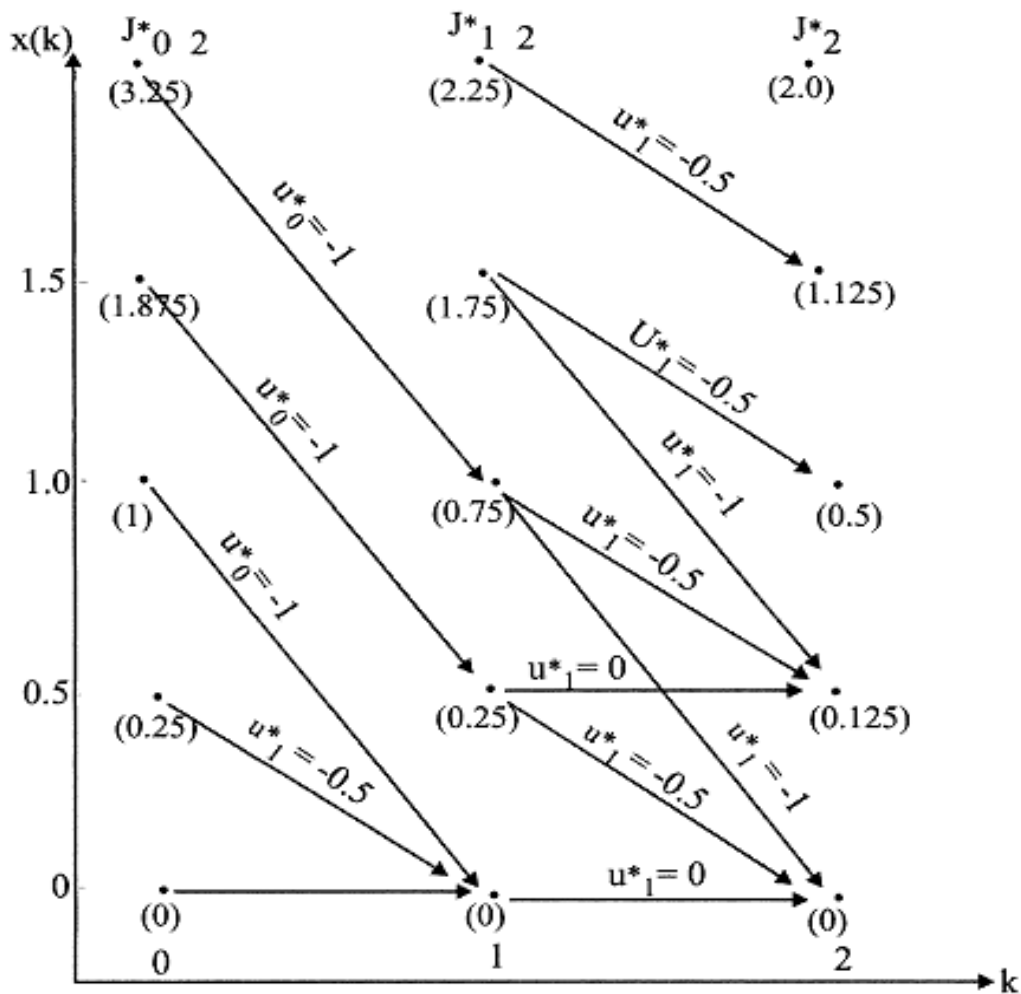


Figure (2.5) Cadre de programmation dynamique de l'Etat optimale

Remarque II.6

Dans l'exemple précédent, il se trouve que pour le contrôle et la quantification donnée, l'état et les valeurs de contrainte (2.20) et (2.21), respectivement, les valeurs calculées en utilisant $x(k + 1) = x(k) + u(k)$ doivent coïncider exactement avec les valeurs quantifiées. Dans certains cas, il peut arriver que pour le contrôle et la quantification donné l'état et les valeurs de contrainte, les valeurs correspondantes des Etats ne coïncident pas exactement avec les valeurs quantifiées, dans ce cas, nous avons besoin pour effectuer une sorte d'interpolation sur les valeurs. Par exemple, disons, la contrainte étatique et de quantification :

$$\begin{cases} -1 \leq x(k) \leq +2, & k = 0, 1 \text{ où} \\ x(k) = -1, 0, 0.5, 1 & 2 \end{cases} \quad (2.26)$$

Chapitre II

Commande optimale des systèmes dynamiques

Alors, pour $x(1) = 2$ et $u(1) = -0.5$, quand nous essayons d'utiliser l'équation d'état (2.18) pour trouver le $x(2) = x(1) + u(1)$, nous obtenons $x(2) = 1.5$, qui n'est pas une valeur admissible quantifiée, est dans la contrainte (limite). Par conséquent, nous ne pouvons pas tout simplement calculer la quantité $J_2 = 0.5x^2(2) = 0.5(1.5)^2$, au lieu d'utiliser l'interpolation, nous la calculons comme suit :

$$\begin{aligned} J_2 &= 0.5(x(2) = 1.5)^2 \\ &= 0.5(x(2) = 1) + \frac{0.5(x(2) = 2)^2 - 0.5(x(2) = 1)^2}{2} \\ &= 0.5 + \frac{2-0.5}{2} = 1.25 \end{aligned} \tag{2.27}$$

Nous remarquons que la technique de programmation dynamique est une méthode qui nécessite un calcul intensif en particulier avec l'augmentation de l'ordre et le nombre d'étapes du système. Toutefois, depuis Bellman, les avancées énormes à grande vitesse des outils de calcul se sont avérées très efficaces pour régler les problèmes de dimensionnalité et de calcul inhérent de la programmation dynamique, ces problèmes peuvent être une aubaine en raison des immenses avantages particuliers que peut apporter la programmation dynamique dans le traitement à la fois des systèmes linéaire et non linéaires et dans le traitement des contraintes sur les états et les contrôles.

II.6 Equation de Hamilton-Jacobi-Bellman [8]

Dans cette partie, nous présentons une autre méthode d'obtention de la commande optimale en boucle fermée, cette méthode consiste à utiliser le principe d'optimalité et l'équation de Hamilton-Jacobi-Bellman (HJB), le principe dit simplement que toute partie de valeur de trajectoire optimale est optimale. Alternativement, Considérons le système de contrôle général :

$$\dot{x}(t) = f(x(t), u(t), t) \tag{2.28}$$

Et la fonction coût (index de performance) est :

$$J(x(t_0), t_0) = \int_{t_0}^{t_f} V(x(t), u(t), t) dt \tag{2.29}$$

Maintenant, nous fournissons l'approche alternative, appelée approche de Hamilton-Jacobi de Bellman et nous obtenons une loi de commande en fonction des variables d'état, conduisant à la commande optimale en boucle fermée. Ceci est important à partir du point de vue pratique dans la mise en œuvre du contrôle optimal.

Chapitre II

Commande optimale des systèmes dynamiques

Nous allons définir une fonction scalaire $J^*(x^*(t), t)$ comme la valeur minimale de l'indice de performance (fonction coût) J , pour un état initial $x^*(t)$ au temps t :

$$J^*(x^*(t), t) = \int_t^{t_f} V(x^*(\tau), u^*(\tau), \tau) d\tau \quad (2.30)$$

En d'autres termes, $J^*(x^*(t), t)$ est la valeur de l'indice de performance évalué le long de la trajectoire optimale à partir de $x(t)$. Ici, nous avons utilisé le principe d'optimalité de dire que la trajectoire de t à t_f est optimale. Cependant, nous ne sommes pas intéressés à trouver la commande optimale pour un état initial particulier $x(t)$, mais pour toutes les conditions initiales non précisées. Ainsi, l'intérêt est en $J(x_{t_0}, t_0)$ comme une fonction de x_{t_0} .
Considérons maintenant :

$$\begin{aligned} \frac{dJ^*(x^*(t), t)}{dt} &= \left(\frac{\partial J^*(x^*(t), t)}{\partial x^*} \right)' \dot{x}^*(t) + \frac{\partial J^*(x^*(t), t)}{\partial t} \\ &= \left(\frac{\partial J^*(x^*(t), t)}{\partial x^*} \right)' f(x^*(t), u^*(t), t) + \frac{\partial J^*(x^*(t), t)}{\partial t} \end{aligned} \quad (2.31)$$

De (2.30), nous avons

$$\left(\frac{dJ^*(x^*(t), t)}{dt} \right) = -V(x^*(t), u^*(t), t) \quad (2.32)$$

en utilisant (2.31) et (2.32) on aura :

$$\begin{aligned} \left(\frac{\partial J^*(x^*(t), t)}{\partial t} \right) + V(x^*(t), u^*(t), t) \\ + \left(\frac{\partial J^*(x^*(t), t)}{\partial x^*} \right)' f(x^*(t), u^*(t), t) = 0 \end{aligned} \quad (2.33)$$

l'Hamiltonien est définie comme suit

$$H = V(x(t), u(t), t) + \left(\frac{\partial J^*(x^*(t), t)}{\partial x^*} \right)' f(x(t), u(t), t) \quad (2.34)$$

On utilisant (2.34) dans (2.33) on aura :

$$\boxed{\left(\frac{\partial J^*(x^*(t), t)}{\partial t} \right) + H \left(x^*(t), \left(\frac{\partial J^*(x^*(t), t)}{\partial x^*} \right)', u^*(t), t \right) = 0; \forall t \in [t_0, t_f]} \quad (2.35)$$

Chapitre II

Commande optimale des systèmes dynamiques

avec la condition limite de (2.30) :

$$J^*(x^*(t_f), t_f) = 0 \quad (2.36)$$

Où

$$J^*(x^*(t_f), t_f) = S(x^*(t_f), t_f) \quad (2.37)$$

Si l'index de performance de (2.29) contient une fonction de coût terminal. L'équation (2.35) est appelée équation de Hamilton-Jacobi. Elle est aussi appelée l'équation d'Hamilton-Jacobi de Bellman (HJB).

Considérons la fonction $\lambda^*(t)$ donnée par la relation

$$\lambda^*(t) = \frac{\partial J^*(x^*(t), t)}{\partial x^*} \quad (2.38)$$

Et supposons que :

$$\dot{\lambda}^*(t) = - \left(\frac{\partial H}{\partial x} \right)_* \quad (2.39)$$

Le contrôle optimal $u^*(t)$ est obtenu à partir de :

$$\left(\frac{\partial H}{\partial u} \right)_* = 0 \rightarrow u^*(t) = h(x^*(t), J_x^*, t) \quad (2.40)$$

Et si nous comparons (2.38) et (2.39), nous obtenons

$$\frac{d}{dt} \left(\frac{\partial J^*(x^*(t), t)}{\partial x^*} \right) = \frac{d}{dt} [\lambda^*(t)] = - \frac{\partial H(x^*(t), \left(\frac{\partial J^*(x^*(t), t)}{\partial x} \right), u^*(t), t)}{\partial x^*} \quad (2.41)$$

En utilisant :

$$J_t^* = \frac{\partial J^*(x^*(t), t)}{\partial t}; \quad J_x^* = \frac{\partial J^*(x^*(t), t)}{\partial x^*} \quad (2.42)$$

L'équation HJB (2.35) devient

$$\boxed{J_t^* + H(x^*(t), J_x^*, u^*(t), t) = 0.} \quad (2.43)$$

En général, cette équation, est une équation différentielle non linéaire, qui peut être résolue, Une fois que J^* est connu, son gradient J_x^* peut être calculé et le contrôle optimal $u^*(t)$ est obtenu à partir de (2.40), dans la plus part des cas, la solution de l'équation HJB est très difficile.

La procédure de résolution est résumée dans le tableau (2.3)

Table (2.3) Résumé de l'approche de la procédure de Hamilton-Jacobi-Bellman (HJB)

A. Enoncé du Problème	
système de contrôle général $\dot{x}(t) = f(x(t), u(t), t),$ Indice de performance (fonction coût) $J = S(x(t_f), t_f) + \int_{t_0}^{t_f} V(x(t), u(t), t) dt,$ et les conditions aux limites : $x(t_0) = x_0$ Trouver le contrôle optimal	
B. solution du Problème	
Etape 1	Former la fonction H $H(x(t), u(t), J_x^*, t) = V(x(t), u(t), t) + J_x^{*'} f(x(t), u(t), t).$
Etape 2	Minimiser H $\left(\frac{\partial H}{\partial u}\right)_* = 0$ et obtenir $u^*(t) = h(x^*(t), J_x^*, t).$
Etape 3	utiliser le résultat de l'étape2, trouver la fonction optimale H^* $H^*(x^*(t), h(x^*(t), J_x^*, t), J_x^*, t) = H^*(x^*(t), J_x^*, t)$ Et obtenir l'équation de HJB.
Etape 4	Résoudre l'équation de HJB $J_t^* + H(x^*(t), J_x^*, t) = 0.$ Avec les conditions limites $J^*(x^*(t_f), t_f) = S(x(t_f), t_f).$
Etape 5	Utilisez la solution J^* , de l'étape 4 afin d'évaluer J_x^* et substituer l'expression de $u^*(t)$ de l'étape 2, pour obtenir le contrôle optimal.

Chapitre II

Commande optimale des systèmes dynamiques

Procédure HJB en utilisant un simple système de premier ordre.

Exemple [8]

Soit un système du premier ordre donné comme suit :

$$\dot{x}(t) = -2x(t) + u(t) \quad (2.44)$$

Et la fonction coût

$$J = \frac{1}{2}x^2(t_f) + \frac{1}{2}\int_0^{t_f}[x^2(t) + u^2(t)] dt \quad (2.45)$$

Trouver la commande optimale du système.

Solution : Tout d'abord, en comparant l'équation (2.44) et la fonction coût (2.45) à la formulation générale de l'équation (2.28) et la fonction coût (2.29) respectivement, nous obtenons les égalités suivantes :

$$V(x(t), u(t), t) = \frac{1}{2}u^2(t) + \frac{1}{2}x^2(t); \quad S(x(t_f), t_f) = \frac{1}{2}x^2(t_f)$$

$$f(x(t), u(t), t) = -2x(t) + u(t) \quad (2.46)$$

Maintenant, nous utilisons la procédure résumée dans le tableau (2.3)

- **Etape 1 :** l'Hamiltonien (2.34) est

$$\begin{aligned} H(x^*(t), J_x, u^*, t) &= V(x(t), u(t), t) + J_x f(x(t), u(t), t) \\ &= \frac{1}{2}u^2(t) + \frac{1}{2}x^2(t) + J_x(-2x(t) + u(t)). \end{aligned} \quad (2.47)$$

- **Etape 2:** Pour un contrôle sans contrainte, une condition nécessaire pour l'optimisation est

$$\frac{\partial H}{\partial u} = 0 \rightarrow u(t) + J_x = 0 \quad (2.48)$$

Et en résolvant

$$u^*(t) = -J_x. \quad (2.49)$$

- **Etape 3** utiliser la commande optimale (2.49) et (2.47) et former la fonction optimale

$$H^* = \frac{1}{2}(-J_x)^2 + \frac{1}{2}x^2(t) + J_x(-2x(t) - J_x) = -\frac{1}{2}J_x^2 + \frac{1}{2}x^2(t) - 2x(t)J_x \quad (2.50)$$

Maintenant, en utilisant les relations précédentes, l'équation HJB (2.43) devient :

$$J_t - \frac{1}{2}J_x^2 + \frac{1}{2}x^2(t) - 2x(t)J_x = 0 \quad (2.51)$$

Avec la condition limite de (2.37)

Chapitre II

Commande optimale des systèmes dynamiques

$$J(x(t_f), t_f) = S(x(t_f), t_f) = \frac{1}{2}x^2(t_f). \quad (2.52)$$

- **Étape 4** Une façon de résoudre l'équation HJB (2.51) avec la condition limite (2.52) est d'assumer une solution et vérifier si elle satisfait l'équation.

Puisque nous voulons le contrôle optimal (2.49) en termes des états et la fonction coût est une fonction quadratique des Etats et des contrôles, on peut deviner la solution suivante

$$J(x(t)) = \frac{1}{2}p(t)x^2(t) \quad (2.53)$$

Où, $p(t)$ est la fonction inconnue à déterminer, et possède comme condition limite

$$J(x(t_f)) = \frac{1}{2}x^2(t_f) = \frac{1}{2}p(t_f)x^2(t_f) \quad (2.54)$$

Ce qui nous donne

$$p(t_f) = 1 \quad (2.55)$$

Ensuite, en utilisant (2.53), nous obtenons

$$J_x = p(t)x(t); \quad J_t = \frac{1}{2}\dot{p}(t)x^2(t), \quad (2.56)$$

Conduisant à la commande optimale en boucle fermée de (2.49)

$$u^*(t) = -p(t)x^*(t) \quad (2.57)$$

En utilisant la commande optimale (2.56) dans l'équation HJB (2.51), on aura

$$\left(\frac{1}{2}\dot{p}(t) - \frac{1}{2}p^2(t) - 2p(t) + \frac{1}{2}\right)x^{*2}(t) = 0 \quad (2.58)$$

Pour tout $x^*(t)$, la relation précédente devient

$$\frac{1}{2}\dot{p}(t) - \frac{1}{2}p^2(t) - 2p(t) + \frac{1}{2} = 0 \quad (2.59)$$

qui, lors de la résolution avec la condition aux limites (2.55) devient

$$p(t) = \frac{(\sqrt{5} - 2) + (\sqrt{5} + 2) \left[\frac{3 - \sqrt{5}}{3 + \sqrt{5}} \right] e^{2\sqrt{5}(t-t_f)}}{1 - \left[\frac{3 - \sqrt{5}}{3 + \sqrt{5}} \right] e^{2\sqrt{5}(t-t_f)}} \quad (2.60)$$

- **Étape 5:** En utilisant la relation (2.60), nous avons le contrôle optimal en boucle fermée de (2.57).

Chapitre II

Commande optimale des systèmes dynamiques

Remarque II.6 Notons que, comme $t_f \rightarrow \infty$, $p(t)$ dans (2.60) devient $p(\infty) = \bar{p} = \sqrt{5} - 2$, et le contrôle optimal (2.56) est

$$u(t) = -(\sqrt{5} - 2)x(t) \quad (2.61)$$

Conclusion

Dans ce chapitre nous avons vu deux façons de résolution d'un problème de commande optimale. Le premier est le principe du maximum de Pontryaguine. Ce dernier peut être appliqué sur des problèmes d'optimisation comportant des conditions spécifiques et techniques variées (frontières, temps final libre, contraintes...etc). Il permet de transformer un problème d'optimisation global dans le temps en un problème local [14].

Le second principe est celui de Bellman, ce dernier a été bien expliqué dans ce chapitre, il s'applique sur tout système dynamique muni d'un critère additif et conduit à la programmation dynamique sur des équations de récurrence. Il est utilisé principalement sur des problèmes formulés en temps discret [14].

Introduction

La commande optimale des systèmes hybrides est un sujet très important. Ce problème a été étudié en utilisant le principe de maximum [11,12,13], ou la programmation dynamique [8] (voir chapitre II). Les équations de Hamilton-Jacobi-Bellman en tant qu'un Système d'Inégalités Quasi Variationelles (SQVIs) sont établies. Pour les systèmes non hybrides, il a été montré que la fonction coût est la solution de viscosité unique de ce SQVIs [13]. Pour les systèmes hybrides des techniques basées sur la solution discontinue de viscosité ont été utilisées. Ainsi, le développement de méthodes numériques efficaces pour la résolution des équations de HJB est un élément clé dans l'application de l'approche de la programmation dynamique à la solution de problèmes de commande optimale de systèmes hybrides.

Pour les systèmes non hybrides, la plus grande partie de méthodes numériques utilisées appartiennent à l'approche de l'approximation d'une chaîne de Markov[16]. Ces méthodes discrétisent l'espace d'état, la commande et le temps, ensuite procèdent au calcul de la solution approximée des équations de HJB.

Les équations de HJB constituent un système de QVIs couplées entre eux par un opérateur non linéaire. Dans ce chapitre nous allons implémenter un algorithme basé sur le découplage de ce système de QVIs ensuite l'application de méthodes de discrétisation pour le calcul de la solution. Le programme sera appliqué à deux exemple.

Nous allons commencer par une représentation théorique de la méthode.

III.1 Commande optimale de systèmes à commutation

Considérons un système à commutation qui consiste en un ensemble de sous-systèmes à temps continu et ayant un nombre fini d'états continus décrit par :

$$\dot{x} = f(x(t), q, u(t))$$

Ici $x(t) \in R^n$ est l'état continu à l'instant t, $u(t) \in U \subset R^m$ est l'entrée de commande continue à t, où U est supposé être compact, $q \in Q = \{q_1, q_2, q_3, \dots, q_L\}$ est un ensemble fini d'indices des sous-systèmes. L'existence et l'unicité des solutions des sous systèmes sont garanties sous la supposition suivante :

Chapitre III Programmation dynamique et résultats de simulation

Supposition III.1

Il existe $L_f > 0$ tel que : $|f(x, q, u)| \leq L_f, \forall (x, q, u) \in (R^n \times Q \times U)$ (3.2)

Et $|f(x, q, u) - f(y, q, u)| \leq L_f |x - y|, \forall x, y \in R^n, q \in Q, u \in U$ (3.3)

A n'importe quel instant, l'un des L sous systèmes est actif et les autres sont inactifs. A des instants particuliers de commutation, le sous système actif est commuté à un autre sous système. Les commutations, sur un intervalle $[0, t_f]$, sont désignées par la séquence finie de commutations suivante :

$$\sigma = \{(\tau_1, q_1), (\tau_2, q_2), \dots, (\tau_K, q_K)\} \text{ où } 0 \leq \tau_1 \leq \tau_2 \leq \dots \leq \tau_K \leq t_f \quad (3.4)$$

Avec $0 \leq \tau_1 \leq \tau_2 \leq \dots \leq \tau_K \leq t_f$ est la séquence des instants de commutation et (q_1, q_2, \dots, q_K) est la séquence de modes de commutation. (τ_k, q_k) signifie que le système commute au sous système d'indice q_k à l'instant τ_k . Ainsi cette séquence satisfait $q_k \in Q, 1 \leq k \leq K$ et $q_{k-1} \neq q_k, 2 \leq k \leq K$.

Dénotons par σ^q une séquence telle que $q_1 = q$. Pour un système à commutation, le signal de commande prend la forme suivante :

$$v = (u, \sigma) \quad (3.5)$$

Où $u \in [0, t_f] \rightarrow U$ est l'entrée de commande continue, qui est supposée être mesurable.

$$u \in \mathcal{U} = \{u: [0, t_f] \rightarrow U | u(\cdot) \text{ mesurable}\} \quad (3.6)$$

Dénotons l'ensemble de tels commande à commutation par $\mathcal{V}^q = \{v^q | v^q = (u, \sigma^q)\}$. Une trajectoire d'état du système à commutation est déterminée par un état initial $x(0)$, un sous système initial q_0 et un signal de commande $v \in \mathcal{V}^q$. $x(\cdot)$ est absolument continu et $q(\cdot)$ est constant par morceau.

Maintenant nous allons formuler le problème de commande optimale de tel système. Pour ceci, nous considérons trois types de coûts associés à une trajectoire d'état d'un système à commutation :

Chapitre III Programmation dynamique et résultats de simulation

Le coût d'évolution continue (ou le coût continu) : $l : R^n \times Q \times U \rightarrow R$.

Le coût de commutation : $\kappa : Q \times Q \rightarrow R^+$.

Le coût terminal : $\psi : R^n \times Q \rightarrow R$.

Le coût continu $l(., q, .)$ est une pénalité sur la trajectoire d'état continu et sur le signal de commande continue correspondant à un sous système $q \in Q$. On suppose que :

Supposition III.2

l est bornée et Lipschitz continu sur x , c'est-à-dire il existe $L_l > 0$ tel que :

$$|l(x, q, u)| \leq L_l, \forall (x, q, u) \in R^n \times Q \times U \quad (3.7)$$

Et : $|l(x, q, u) - l(y, q, u)| \leq L_l |x - y|, \forall x, y \in R^n, q \in Q, u \in U \quad (3.8)$

Le coût de commutation $\kappa(q_1, q_2)$ est le coût de commutation du sous système q_1 au sous système q_2 . Nous supposons que la fonction κ vérifie une inégalité triangulaire donnée par :

Supposition III.3

$$\begin{aligned} \kappa(q, \hat{q}) > 0, \quad \kappa(q, q) = 0, \quad \kappa(q, \hat{q}) < \kappa(q, \tilde{q}) + \kappa(\tilde{q}, \hat{q}); \\ \text{pour tout } q, \hat{q}, \tilde{q} \in Q, q \neq \hat{q} \neq \tilde{q} \end{aligned} \quad (3.9)$$

Pour le coût final ψ on supposera qu'il est Lipschitz continu et borné.

Maintenant le coût optimal attaché à la trajectoire d'état sur l'intervalle $[0, t_f]$ avec l'état initial $x(0) = x$, le sous système initial $q_0 = q \in Q$ et le signal de commande à commutations $v \in \mathcal{V}^q$ est donné par :

$$J(x, q, v) = \int_0^{t_f} l(x(t), q(t), u(t)) dt + \sum_{k=1}^K \kappa(q_{k-1}, q_k) + \psi(x(t_f), q_K) \quad (3.10)$$

L'objectif de la commande optimale est : Pour tout $(x, q) \in R^n \times Q$ trouver la commande à commutation $v^* \in \mathcal{V}^q$ telle que :

$$J(x, q, v^*) \leq J(x, q, v), \forall v \in \mathcal{V}^q \quad (3.11)$$

Chapitre III Programmation dynamique et résultats de simulation

A la différence avec d'autres approches [11,12,13], le coût contient une partie positive qui est due seulement aux commutations. Il a été montré que pour de telle fonction de coût il y aura au plus un nombre fini de commutations dans une commande optimale ou sous optimale. On verra que ce fait joue un rôle clé dans le calcul du coût optimal.

III.2 Système d'Inégalités Quasi Variationnelles de la commande optimale à commutations

Maintenant considérons un problème de commande optimale à commutation sur un intervalle $[t, t_f]$. On peut, de manière similaire, définir le coût total $J(x, q, v, t)$ pour $v \in \mathcal{V}_t^q$ sur cet intervalle. On applique l'approche de la programmation dynamique pour trouver le coût optimal dénoté comme suit :

$$V(x, q, t) = \inf_{v \in \mathcal{V}_t^q} J(t, x, q, v) \quad (3.12)$$

Dans ce chapitre, nous allons implémenter une méthode pour le calcul de cette fonction et ensuite, on calcul le contrôleur optimal à partir de ses valeurs. Nous commencerons par donner quelques propriétés de cette fonction.

Proposition III.1

La fonction du coût optimal est bornée :

$$V(x, q, t) \leq L_l(t_f - t) + L_\omega, (x, q, t) \in R^n \times Q \times [0, t_f] \quad (3.13)$$

Et elle est localement Lipschitz continue sur $R^n \times [0, t_f]$ pour tout $q \in Q$.

Corollaire III.1

Soit $v^\varepsilon = v^\varepsilon(x, q, t) \in \mathcal{V}_t^q$ une commande à commutation ε -optimale pour (x, q, t) c'est-à-dire :

$$J(x, q, t, v) < V(x, q, t) + \varepsilon \quad (3.14)$$

Et soit $N(v^\varepsilon) = N(v^\varepsilon(x, q, t))$ le nombre total de commutations pour v^ε , alors $N(v^\varepsilon)$ est uniformément borné par :

$$N(v^\varepsilon(x, q, t)) \leq \frac{2(L_f t_f + L_\psi) + \varepsilon}{\tilde{\kappa}}, \forall (x, q, t) \in R^n \times Q \times [0, t_f] \quad (3.15)$$

Chapitre III Programmation dynamique et résultats de simulation

Où $\tilde{\kappa} = \min_{q, \tilde{q}, q \neq \tilde{q}} \kappa(q, \tilde{q}) > 0$

Proposition III.2 (Principe de la programmation dynamique) :

La fonction du coût optimale satisfait :

$$1. \text{ La condition terminale : } V(x, q, t_f) = \min \left\{ \min_{\tilde{q} \neq q} (\kappa(q, \tilde{q}) + V(x, \tilde{q}, t_f)), \psi(x, q) \right\} \quad (3.16)$$

2. Pour tout $x \in R^n$ et $0 \leq t \leq t_f$

$$V(x, q, t) \leq \min_{\tilde{q} \neq q} (\kappa(q, \tilde{q}) + V(x, \tilde{q}, t)) \quad (3.17)$$

Et pour tout petit $\delta > 0$ tel que $\delta + t < t_f$ et tout $u \in \mathcal{U}$:

$$V(x, q, t) \leq \int_t^{t+\delta} l(x(s), q, u(s)) ds + V(x(t+\delta), q, t+\delta) \quad (3.18)$$

3. Si pour un $x \in R^n$ et $t < t_f$, l'inégalité (3.17) est strict, alors il existe un $\delta_0 > 0$ avec $\delta_0 + t < t_f$, on a pour tout $\delta < \delta_0$:

$$V(x, q, t) \geq \inf_{u \in \mathcal{U}} \left\{ \int_t^{t+\delta} l(x(s), q, u(s)) ds + V(x(t+\delta), q, t+\delta) \right\} \quad (3.19)$$

Maintenant nous introduisons l'opérateur suivant :

$$(\mathcal{M}V)(x, q, t) = \min_{\tilde{q} \neq q} (\kappa(q, \tilde{q}) + V(x, \tilde{q}, t)), \forall V \in C(R^n \times Q \times [0, t_f]) \quad (3.20)$$

Et aussi le Hamiltonien :

$$\mathcal{H}(x, q, p) = \sup_{u \in U} \{-l(x, q, u) - p \cdot f(x, q, u)\}, \forall (x, q, p) \in R^n \times Q \times R^n \quad (3.21)$$

La fonction du coût optimal $V(x, q, t)$ est caractérisée par la solution des équations de HJB. Le théorème suivant formule ce résultat.

Chapitre III Programmation dynamique et résultats de simulation

Théorème III.1

La fonction de coût optimal $V(x, q, t)$ est la solution unique de viscosité du système d'équations aux dérivées partielles suivant :

$$\begin{cases} \max \left\{ V(x, q, t) - (\mathcal{M}V)(x, q, t), -\frac{\partial V}{\partial t}(x, q, t) + H(x, q, \nabla_x V(x, q, t)) \right\} = 0 \\ V(x, q, t_f) = \min \{ (\mathcal{M}V)(x, q, t_f), \psi(x, q) \} \end{cases} \quad (3.22)$$

Maintenant supposons qu'on a trouvé la solution du système d'équations aux dérivées partielles (3.22), avec une solution lisse V , alors le signal de commande à commutation peut être construit à partir de cette solution. La commande est composée de deux parties. La première est une contre réaction pour chaque sous système sous la forme habituelle, c'est-à-dire une fonction :

$$K_s : R^n \times Q \times [0, t_f] \rightarrow U \quad (3.23)$$

Qui est obtenue à partir de :

$$K_s(x, q, t) = \arg \max_{u \in U} \{ -l(x, q, u) - \nabla_x V(x, q, t) \cdot f(x, q, t) \} \quad (3.24)$$

$K_s(x, q, t)$ transforme le sous système d'indice q en un système continu bouclé autonome :

$$\dot{x}(t) = f(x(t), q, K_s(x(t), q, t)) \quad (3.25)$$

La deuxième partie est la partie discrète de la commande qui a la forme :

$$\Lambda = (\mathcal{O}, r_o) \quad (3.26)$$

Où $\mathcal{O} : [0, t_f] \rightarrow 2^{R^n}$ est l'ensemble de commutations optimales obtenu à partir de :

$$\mathcal{O}(t) = \{ (x, q) \in (R^n \times Q) \mid V(x, q, t) = (\mathcal{M}V)(x, q, t) \} \quad (3.27)$$

Si $x(t) \in \mathcal{O}(t)$ à un instant t , alors l'action optimale est de faire une commutation selon la loi $r_o : [0, t_f] \times R^n \times Q \rightarrow Q$ qui est obtenue par :

$$r_o(t, x, q) = \arg \min_{\tilde{q} \neq q} (\kappa(q, \tilde{q}) + V(x, \tilde{q}, t)) \quad (3.28)$$

Chapitre III Programmation dynamique et résultats de simulation

III.3 Les Inégalités Quasi Variationnelles récursives

Généralement, il est difficile de résoudre les équations de HJB (3.22). Ce n'est pas seulement une seule équation dans R^n . C'est un système d'Inégalités Quasi Variationnelles couplées par l'opérateur non linéaire \mathcal{M} défini par (3.20).

Une autre formulation a été établie qui permet de découpler ce système de SQVIs. Pour montrer comment ceci a été obtenu, nous allons donner une définition qui identifie les différentes classes de commandes à commutations par le nombre maximal de commutations permises :

Définition III.1

Pour tout entier $N \geq 0$, on appelle $\mathcal{V}_t^q(N) \in \mathcal{V}_t^q = \{(u(\cdot), \sigma^q(N))\}$ un signal de commutation de classe N, si la séquence de commutation $\sigma^q = ((\tau_1, q_1), (\tau_2, q_2), \dots, (\tau_K, q_K))$ satisfait : $t \leq \tau_1 \leq \tau_2 \leq \dots \leq \tau_K \leq t_f$ et $q_k \neq q_{k+1}, 1 \leq k \leq K-1$ et la contrainte $K \leq N$.

Un signal de commande de classe N a au plus N commutations. D'où :

$$\mathcal{V}_t^q(N_1) \subset \mathcal{V}_t^q(N_2), \quad N_1 \leq N_2 \quad (3.29)$$

$\mathcal{V}_t^q(0)$ est tout simplement l'ensemble de signaux de commande sans commutations et $\mathcal{V}_t^q(\infty)$ est l'ensemble de signaux de commande permettant un nombre arbitraire de commutation.

Maintenant on peut définir des problèmes de minimisation pour chaque classe de commande :

$$V^N(x, q, t) = \inf_{v \in \mathcal{V}_t^q(N)} J(x, q, t, v) \quad (3.30)$$

De l'équation (3.29) on a :

$$V^{N_1} \geq V^{N_2}, \quad N_1 \leq N_2 \quad (3.31)$$

La plus grande valeur est V^0 qui correspond à une commande sans commutation et la plus petite valeur est V^∞ qui correspond au coût optimale d'origine de l'équation (3.12).

Chapitre III Programmation dynamique et résultats de simulation

Maintenant le corollaire (3.5) stipule que le nombre de commutations dans tout commande

optimale est borné par un nombre entier positif $\tilde{K} = \left\lceil \frac{2(L_f t_f + L_\psi)}{\tilde{\kappa}} \right\rceil$, l'entier le plus petit

juste supérieur à $\frac{2(L_f t_f + L_\psi)}{\tilde{\kappa}}$. Ceci implique que :

$$V^N = V^{\tilde{K}}, \forall N \geq \tilde{K} \quad (3.32)$$

A partir de cette équation on a :

$$V^{\tilde{K}} = V \quad (3.33)$$

Nous allons dans ce qui suit voir comment calculer $V^{\tilde{K}}$ à partir de $V^0, V^1, \dots, V^{\tilde{K}-1}$ de manière récursive en résolvant les équations de HJB pour $V^N, 0 \leq N \leq \tilde{K}$. Nous commençons par le principe de la programmation dynamique pour ces fonctions :

Théorème III.2

Les fonctions de coût optimales $V^N, 0 \leq N \leq \tilde{K}$ satisfont :

1. Conditions terminales : Pour tout $(x, q) \in (R^n \times Q)$

$$V^0(x, q, t_f) = \psi(x, q) \quad (3.34)$$

2. Et pour $0 < N \leq \tilde{K}$,

$$V^N(x, q, t_f) = \min \left\{ (\mathcal{M}V^{N-1})(x, q, t_f), \psi(x, q) \right\} \quad (3.35)$$

3. Pour $(x, q) \in (R^n \times Q), 0 \leq t \leq t_f$ et $\delta > 0$ tel que $t + \delta < t_f$

$$V^0(x, q, t) = \inf_{u \in \mathcal{U}} \left\{ \int_t^{t+\delta} l(x(s), q, u(s)) ds + V^0(x(t+\delta), q, t+\delta) \right\} \quad (3.36)$$

4. Pour $0 \leq t < t_f, 0 \leq N < \tilde{K}$, on a :

$$\text{Pour tout } (x, q) \in (R^n \times Q) \quad V^N(x, q, t) \leq (\mathcal{M}V^{N-1})(x, q, t); \quad (3.37)$$

Chapitre III Programmation dynamique et résultats de simulation

5. Pour $(x, q) \in (R^n \times Q)$, et $\delta > 0$ tel que $t + \delta < t_f$ et tout $u \in U$:

$$V^N(x, q, t) \leq \int_t^{t+\delta} l(x(s), q, u(s)) ds + V^0(x(t+\delta), q, t+\delta); \quad (3.38)$$

6. Si pour un $(x, q) \in (R^n \times Q)$, et $t < t_f$ l'inégalité (3.37) est strict, alors il existe $\delta_0 > 0$ tel que $t + \delta_0 < t_f$, pour tout $\delta < \delta_0$ on a :

$$V^N(x, q, t) \geq \inf_{u \in U} \int_t^{t+\delta} l(x(s), q, u(s)) ds + V^N(x(t+\delta), q, t+\delta); \quad (3.39)$$

Le théorème suivant donne un nouvel ensemble d'équations de HJB devant être satisfaites par les fonctions de coût optimal V^N dans le sens de viscosité :

Théorème III.3

Les fonctions de coût optimal $V^N, 0 \leq N \leq \tilde{K}$ sont la solution unique de viscosité du système :

$$\begin{cases} -\frac{\partial V^0}{\partial t}(x, q, t) + H(x, q, \nabla_x V^0(x, q, t)) = 0 \\ V^0(x, q, t_f) = \psi(x, q) \end{cases} \quad (3.40)$$

Et pour $0 < N \leq \tilde{K}$:

$$\begin{cases} \max \left\{ V^N(x, q, t) - (\mathcal{M}V^{N-1})(x, q, t), -\frac{\partial V^N}{\partial t}(x, q, t) + H(x, q, \nabla_x V^N(x, q, t)) \right\} = 0 \\ V^N(x, q, t_f) = \min \{ (\mathcal{M}V)(x, q, t_f), \psi(x, q) \} \end{cases} \quad (3.41)$$

Le système des SQVIs (3.41) est différent du système (3.22), du fait que pour tout $1 \leq N \leq \tilde{K}$, (3.41) sont L équations de $V^N(., q, .)$ $q \in Q$ indépendantes. La solution pour tout $V^N(., q, .)$ dépend seulement de fonctions connue $V^{N-1}(., q, .), q \in Q$ et elle ne dépend pas des fonctions inconnues $V^N(., \tilde{q}, .)$ $\tilde{q} \in Q$.

L'algorithme pour le calcul des fonctions $V^N(., q, .), 0 \leq N \leq \tilde{K}, q \in Q$ à partir des équations (3.40) et (3.41) est comme suit :

Chapitre III

Programmation dynamique et résultats de simulation

Algorithme III.1

Étape 1 :

Calculer $V^0(., q, .), q \in Q$ selon l'équation (3.40)

Étape 2 :

Pour $N = 1, \dots, \tilde{K}$

Étape 2.1

Calculer $(\mathcal{M}V^{N-1})(., q, .)$ selon :

$$(\mathcal{M}V^{N-1})(., q, .) = \min_{\tilde{q} \neq q} \{ \kappa(q, \tilde{q}) + V^{N-1}(x, \tilde{q}, t) \}$$

Étape 2.2 :

Calculer $V^N(., q, .), q \in Q$ selon l'équation (3.41)

Étape 3 :

Pour une borne supérieure fixe de l'erreur de valeur ε , si $\|V^N - V^{N-1}\|_\infty \leq \varepsilon$, alors $V = V^N$ et le programme arrête, sinon aller à l'étape 2.2 pour le calcul de V^{N+1} jusqu'à $V^{\tilde{K}}$.

Les calculs dans cet algorithme sont concentrés sur la résolution de $\tilde{K} \times \text{card}(Q)$ équations aux dérivées partielles. Ainsi l'efficacité de cet algorithme dépend de \tilde{K} qui est la borne supérieure estimée du nombre de max permis de commutations dans une commande optimale à commutation. Ce nombre ne doit pas être trop conservatif. Ce problème est contourné en laissant cette borne supérieure libre et en surveillant seulement l'erreur sur $\|V^N - V^{N-1}\|_\infty$.

III.4 Implémentation de l'algorithme pour la solution des QVIs

Vu que les calculs dans l'algorithme présenté dans la section précédente sont pour la solution de $(\tilde{K} + 1) \times \text{card}(Q)$ QVIs, nous allons donner l'approche numérique pour le calcul de la solution de ces QVIs. Considérons une QVI de la forme :

Chapitre III Programmation dynamique et résultats de simulation

$$\begin{cases} \max \left\{ V(x, t) - \varphi(x, t), -\frac{\partial V^N}{\partial t}(x, t) + \mathcal{H}(x, \nabla_x V(x, t)) \right\} = 0 \\ V(x, t_f) = \min \left\{ (\mathcal{L}V)(x, t_f), \psi(x) \right\} \end{cases} \quad (3.42)$$

Où $\mathcal{H}(x, p) = \sup_{u \in U} \{-l(x, u) - p \cdot f(x, u)\}, (x, p) \in R^n \times R^n$.

Cette équation est une forme spéciale de l'équation générale de HJB de la commande optimale. Alors, les approches générales pour la résolution de l'équation de HJB comme l'approximation par une chaîne de Markov peuvent être appliquées. Cette dernière définit un problème de commande optimale pour une chaîne de Markov discrète pour approximer le problème d'origine de la commande optimale continue. Cette chaîne de Markov est définie sur un espace de variable d'état et de temps discrétisé. Plusieurs méthodes de discrétisation existent [13,15]. La méthode que nous allons utiliser est donnée dans ce qui suit.

Soit $(R^n)^\delta$ un maillage de l'espace R^n de pas δ , centrée à l'origine. Soit $N_\delta(x)$ pour $x \in (R^n)^\delta$ le voisinage :

$$N_\delta(x) = \left\{ z \in (R^n)^\delta \mid z = x \text{ ou } z = x \pm \delta e_j, j = 1, \dots, n \right\} \quad (3.43)$$

Ici $e_j \in R^n$ dénote le j ième vecteur unité. Soit $U^\delta = U \cap (R^n)^\delta$. On introduit :

$$\lambda^\delta \equiv \sup_{x \in (R^n)^\delta, u \in U^\delta} |f(x, u)|_1 \quad (3.44)$$

Où $|v|_1 = |v_1| + |v_2| + \dots + |v_n|$ est la norme 1 dans R^n .

Dénotons par $\Delta t = \delta / \lambda^\delta$, $t_k = t_f - (K_f - k)\Delta t$, $0 \leq k \leq K_f$, et $(0, t_f)^\delta = (t_0, t_1, \dots, t_k)$.

Nous utilisons les approximations suivantes des différences finies :

$$\begin{aligned} \frac{\partial V}{\partial t}(x, t) &= \frac{V(x, t + \delta / \lambda^\delta) - V(x, t)}{\delta / \lambda^\delta} \\ \frac{\partial V}{\partial x_j}(x, t) &= \frac{V(x + \delta e_j, t + \delta / \lambda^\delta) - V(x, t + \delta / \lambda^\delta)}{\delta}, \text{ si } f_j(x, u) \geq 0 \\ \frac{\partial V}{\partial x_j}(x, t) &= \frac{V(x, t + \delta / \lambda^\delta) - V(x - \delta e_j, t + \delta / \lambda^\delta)}{\delta}, \text{ si } f_j(x, u) < 0 \end{aligned} \quad (3.45)$$

Chapitre III Programmation dynamique et résultats de simulation

Si on remplace dans le système d'équation (3.42), on obtient :

$$\begin{cases} V^\delta(x, t_k) = \min \left\{ \varphi(x, t_k), \min_{u \in U} \left\{ \sum_{z \in N_\delta(x)} p^\delta(x, z, u) V^\delta(z, t_{k+1}) + \frac{\delta}{\lambda^\delta} l(x, u) \right\} \right\} \\ V^\delta(x, t_{K_f}) = \min \{ \varphi(x, t_{K_f}), \psi(x, t_{K_f}) \} \end{cases} \quad (3.46)$$

$$\text{Où : } p^\delta(x, z, u) = \begin{cases} 1 - |f(x, u)|_1 / \lambda^\delta & \text{si } z = x, \\ f_j^\pm(x, u) / \lambda^\delta, & \text{si } z = x \pm \delta e_j, j = 1, \dots, n, \\ 0 & \text{si } z \notin N_\delta(x) \end{cases}$$

Avec $f^+(x, u) = \max\{0, f(x, u)\}$ et $f^-(x, u) = \max\{0, -f(x, u)\}$. Cette équation constitue l'équation de HJB discrète correspondant au problème de commande optimale du système hybride.

Dans l'approximation de la dérivée par les différences finies et aux frontières du maillage, on aura besoins des valeurs de la fonction coût à des points qui sont situés en dehors du maillage. Ces valeurs seront remplacées par celles obtenues par interpolation. La méthode utilisée est décrite par la figure suivante :

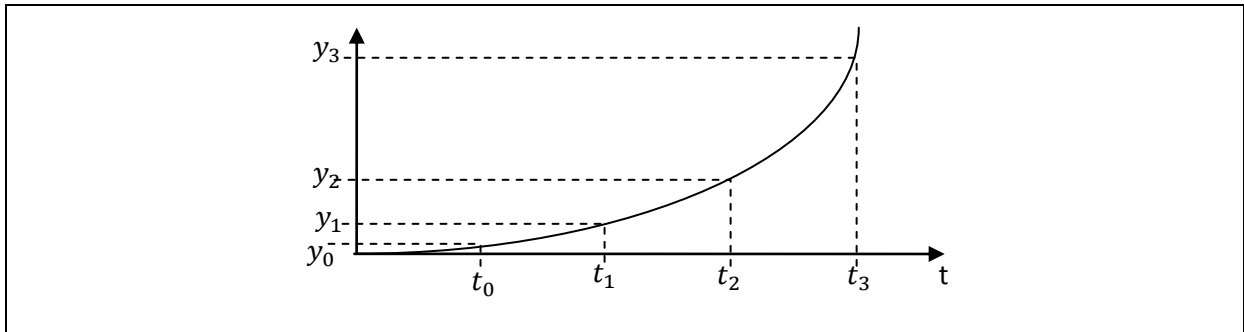


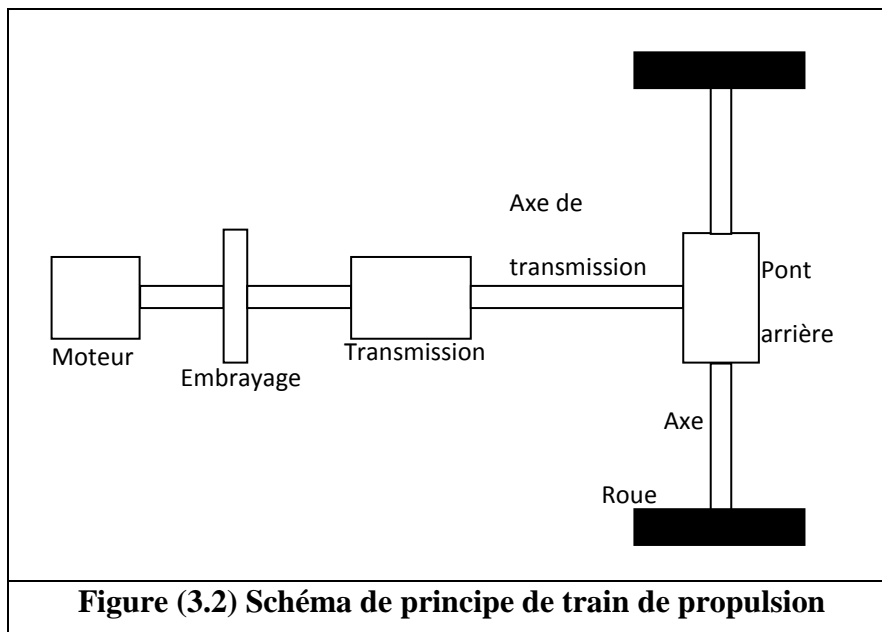
Figure (3.1) La formule de l'interpolation polynomiale parabolique est établie en utilisant le polynôme de Lagrange :

$$\begin{aligned} y_0 &= 3y_1 - 3y_2 + y_3; \text{ pour une interpolation vers } -\infty \\ y_3 &= y_0 - 3y_1 + 3y_2; \text{ pour une interpolation vers } +\infty \end{aligned}$$

Cet algorithme a été appliqué pour la commande automatique de la transmission d'un véhicule. Le modèle du véhicule est établi dans la section suivante.

III.5 Modèle théorique du véhicule [16]

Les parties principales d'un véhicule de poids lourds consistent en un moteur, embrayage, transmission (boîte de vitesse), arbre de transmission, traction finale, l'axe arrière et enfin les roues. Cette ensemble forme la partie connu sous le train de propulsion. Cette partie est fondamentale quand on veut évaluer les dynamiques du véhicule. Elle peut être modélisée par plusieurs façons selon l'objectif et l'utilisation. Nous allons établir le modèle dynamique du véhicule en ce basant sur le schéma de principe de la figure suivante :



Le moteur produit le couple à travers la combustion du diesel mélangé avec un surplus d'air dans une chambre hautement pressurisée. La combustion explosive entraîne le vilebrequin, qui est connecté à son tour à l'embrayage par un arbre. Le couple de sortie du moteur est caractérisé par le couple résultant de la combustion, les frottements internes avec les murs de la chambre de combustion, et le couple externe de l'embrayage. Ainsi, si on a le moment d'inertie, la deuxième loi de Newton donne :

$$J_e \dot{\omega}_e = M_e - M_c \tag{3.47}$$

M_e est le couple produit par le moteur et il est fonction de $M_e = M_e(\omega_e, u)$, avec ω_e est la vitesse de rotation du moteur et u caractérise l'alimentation en carburant. M_c est le couple de l'embrayage.

Chapitre III Programmation dynamique et résultats de simulation

L'embrayage contient deux disques de frottement, qui sont compressés entre eux et connecte le volant moteur avec la transmission. La connexion entre la transmission et l'embrayage est considérée rigide c'est-à-dire :

$$M_t = M_c; \omega_t = \omega_e \quad (3.48)$$

M_t et ω_t , sont respectivement, le couple et la vitesse angulaire à l'entrée de la transmission.

La transmission (boîte de vitesse) consiste en un ensemble d'engrenages connectés de telle sorte que le couple de sortie soit transformé selon la vitesse engagée. Elle est modélisée par un rapport de conversion i_t qui varie selon les caractéristiques de la boîte de vitesse. Une autre caractéristique de la transmission est son rendement η_t ainsi :

$$M_p = i_t \eta_t M_t; \omega_e = i_t \omega_p \quad (3.49)$$

Avec M_p et ω_p sont le couple et la vitesse angulaire de l'arbre de transmission.

L'arbre de transmission connecte la transmission au pont arrière et il est considéré rigide. Le pont arrière est caractérisé par un rapport de transmission i_f et un rendement η_f , en négligeant l'inertie nous obtenons ces relation :

$$M_d = i_f \eta_f M_p = i_f \eta_f M_t; \omega_f = \omega_p = i_f \omega_d \quad (3.50)$$

L'axe arrière connecte le pont arrière aux roues, cette connexion est considérée rigide. La connexion entre les roues et la route est considéré sans glissement et elle est régie par :

$$J_w \dot{\omega}_w = M_w - r_w F_w; v = r_w \omega_w = \frac{r_w \omega_e}{i_f i_t} \quad (3.51)$$

v est la vitesse du véhicule, J_w , r_w , ω_w et F_w sont respectivement l'inertie, le rayon, la vitesse angulaire et la force au niveau de la roue. Si la position du véhicule est dénotée par x_1 alors $v = \dot{x}_1$.

Les équations (3.47-51) permettent d'écrire :

$$M_w = M_d = i_f \eta_f i_t \eta_t M_c; \omega_w = \frac{\omega_e}{i_t i_f}; i_t i_f J_e \dot{\omega}_w = M_e - M_c \quad (3.52)$$

Le couple généré par la moteur est donné par $M_e = h(\omega_e)u$, où la fonction $h(\omega_e)$ caractérise le rendement du moteur.

Chapitre III Programmation dynamique et résultats de simulation

Ces dernières équations donnent :

$$(J_w - (i_t i_f)^2 \eta_f \eta_t J_e) \dot{\omega}_w = i_t i_f \eta_t \eta_f h(\omega_e) u - r_w F_w (3.53)$$

Si on prend en considération les relations entre la vitesse du véhicule et les différentes vitesses angulaire on peut écrire l'équation (3.53) sous la forme :

$$\dot{v} = g(v)u - kv \quad (3.54)$$

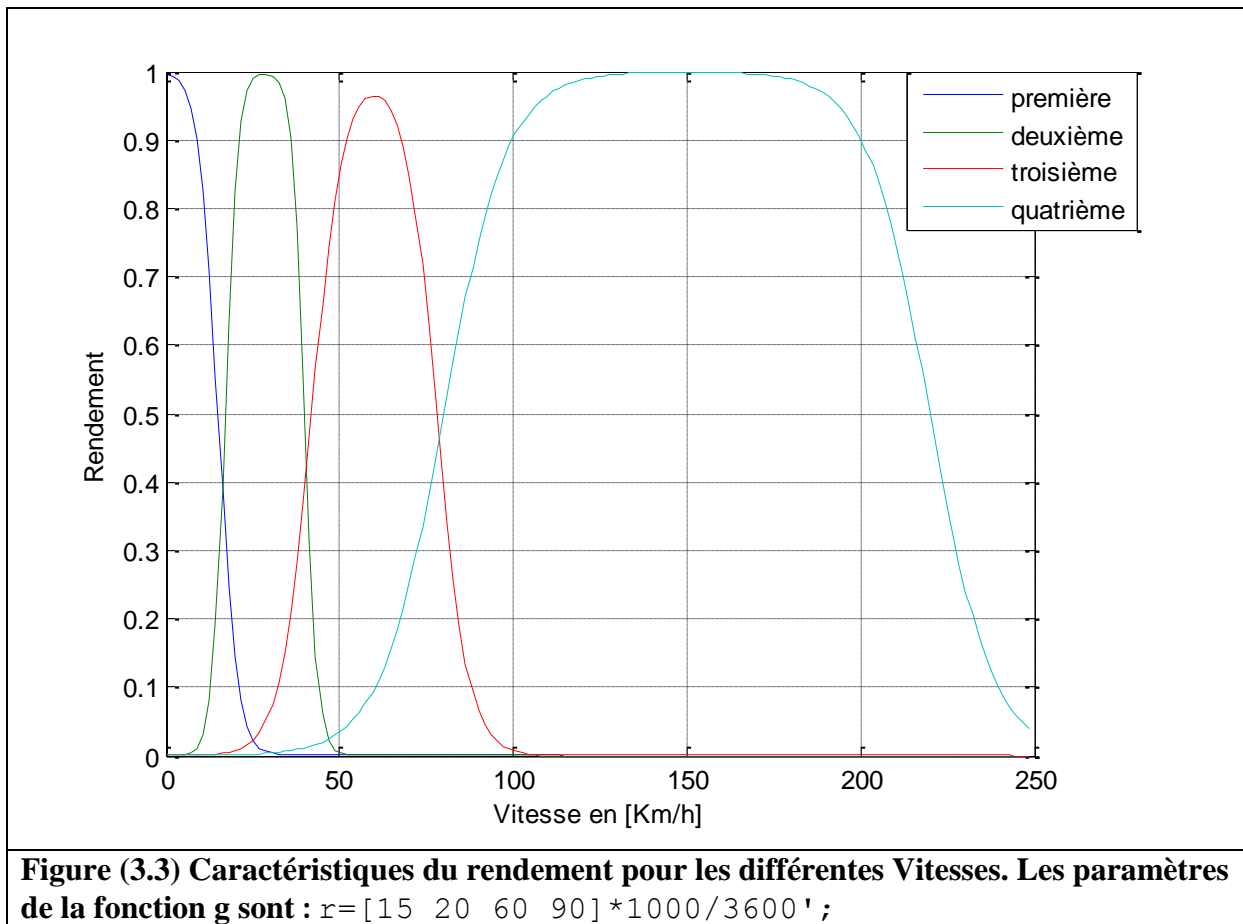
Les équations d'états du véhicule sont donnée par :

$$\begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = g(x_2)u - kx_2 \end{cases} \quad (3.55)$$

La fonction $g(x_2)$ caractérise le rendement du véhicule et u est l'entrée de commande et caractérise l'alimentation en carburant. Elle est déterminée à partir de la position de la pédale de l'accélérateur.

III.6 Résultats de simulation

Pour la simulation de la commande automatique de la transmission du véhicule, nous allons commencer par la détermination de la fonction de rendement du moteur. La fonction $g(x_2)$ varie selon la vitesse engagée, elle est approximée en utilisant la fonction tangente hyperbolique pour obtenir quatre courbes (une pour chaque vitesse), ces courbes sont maximales (à 1) pour les valeurs de vitesse qui entrent dans l'intervalle de rendement optimal de la vitesse engagée (en court) . Les caractéristiques de rendement des différentes vitesses que nous avons adoptées sont données par la figure suivante :

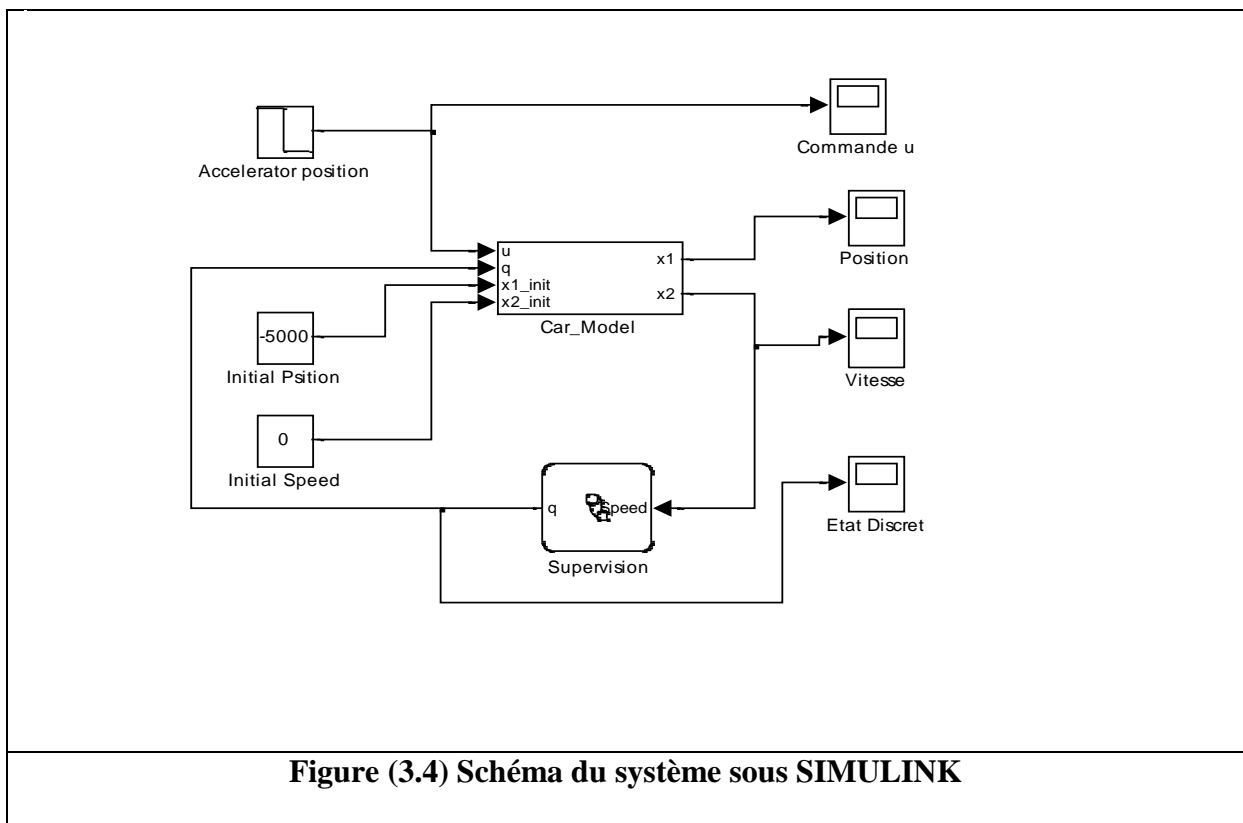


Chapitre III Programmation dynamique et résultats de simulation

Pour avoir une idée précise sur le système qu'on va commander, nous avons simulé ce dernier dans l'environnement SIMULINK avec STATEFLAW.

La supervision de l'état discret est effectuée par le graphe d'état suivant. Selon la vitesse courante renvoyée par SIMULINK, qui modélise la partie continue du système, la partie supervision choisit l'état discret qui permet un rendement maximal.

Nous avons fait la simulation du système pour une distance traversée de 10Km. La commande $u(t)$ varie de +1 à -1 au milieu du trajet. Les résultats obtenus sont représentés par la même figure. Au début, la vitesse du véhicule augmente graduellement, et l'état discret varie de 1, 2, 3, à 4 de telle sorte à suivre l'augmentation de la vitesse. Ensuite, le véhicule continue son déplacement avec une grande vitesse et l'état discret étant à 4 (quatrième vitesse). Quand le véhicule s'approche de sa destination la vitesse se met à diminuer graduellement et l'état discret passe de 4, 3, 2, et enfin à 1.



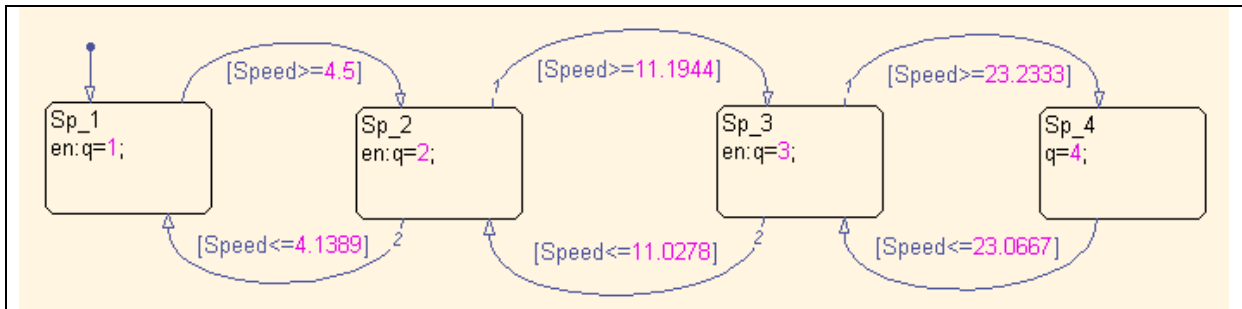


Figure (3.5) Graphe d'état (Flow Chart) de la supervision. Ici la vitesse ayant le plus grand rendement est choisie. Les conditions de transition sont conformes à la caractéristique du rendement avec une hystérésis de +/-0.3611m/sec.

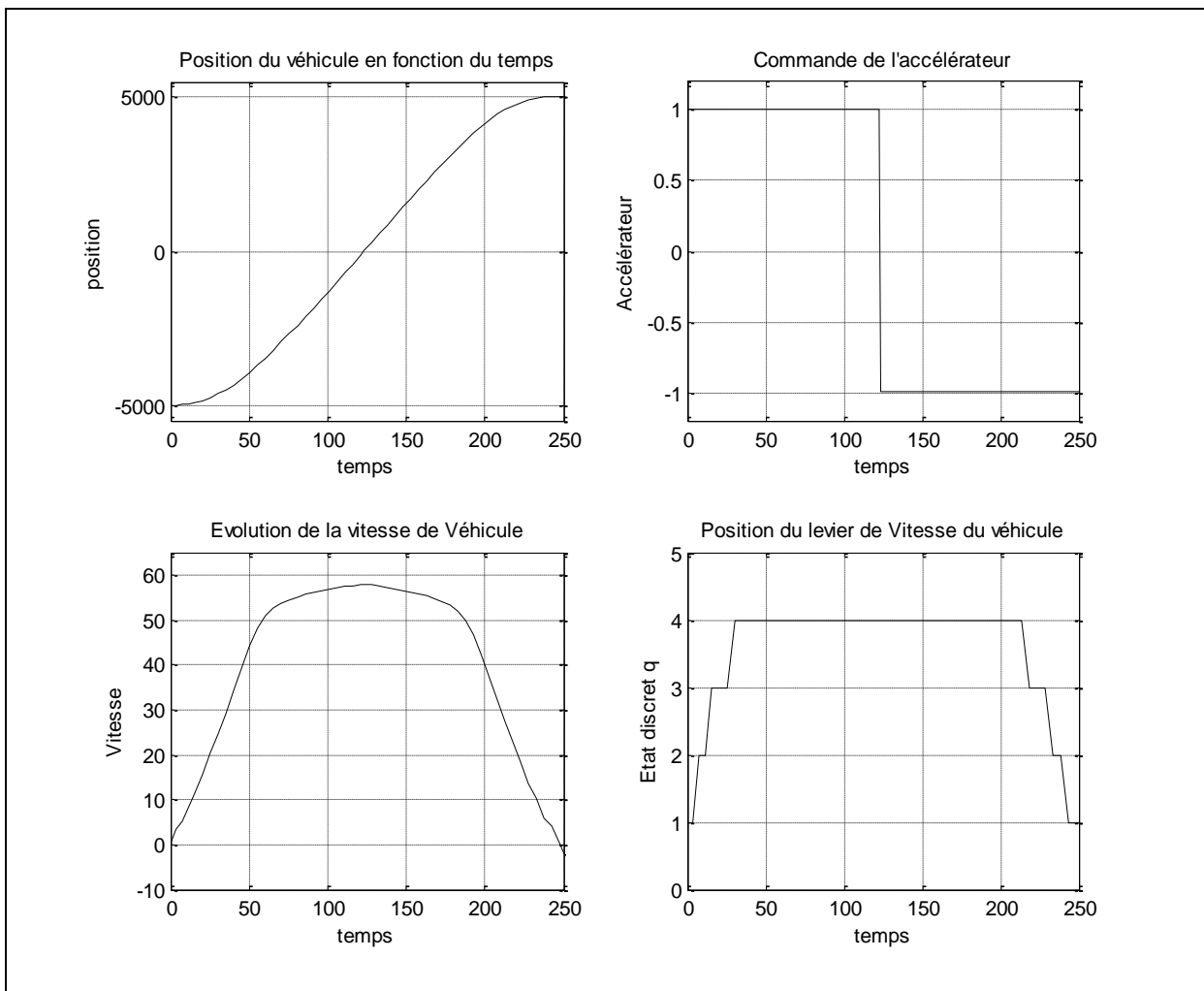


Figure (3.6) Résultat obtenue en utilisant une commande avec supervision discrète (logique). La commande u varie de 1 à -0.99126

Chapitre III Programmation dynamique et résultats de simulation

Les résultats obtenus ici ne sont pas complètement optimaux, car les instants de commutation détectés ici dépendent de la méthode de l'intégration du modèle. Celle-ci engendre des erreurs cumulatives, et vu que le comportement du système est très sensible aux erreurs de calcul de ces instants de commutation, le résultat obtenu n'est que approximatif. Néanmoins, ces courbes nous permettent d'avoir une idée sur les intervalles qu'on doit utiliser pour les différentes variables (position et vitesse), ce qui permet de définir l'espace dans lequel la solution de l'équation de HJB sera calculée. Cet espace doit être centré autour de l'origine.

Pour une distance traversée de 10Km, on prend pour la position l'intervalle $-5000m \leq x_1 \leq 5000m$. D'après la figure la vitesse peut aller jusqu'à 60m/sec, on prend alors l'intervalle $-60m/sec \leq x_2 \leq 60m/sec$. La commande u doit vérifier $-1 \leq u \leq 1$. Pour l'inertie, on prendra $k=0.01$.

Pour normaliser les variables et avoir un maillage centré autour de l'origine, on procédera à une mise à l'échelle en faisant un changement de variable comme suit :

$$z_1 = \frac{x_1}{5000} \text{ et } z_2 = \frac{x_2}{60}$$

Ceci transforme le modèle sous la forme :

$$\begin{cases} \dot{z}_1 = \frac{\dot{x}_1}{5000} = \frac{x_2}{5000} = \frac{60z_2}{5000} = 0.012z_2 \\ \dot{z}_2 = \frac{\dot{x}_2}{60} = \frac{g(x_2)u}{60} - \frac{kx_2}{60} = \frac{g(60z_2)u}{60} - kz_2 \end{cases}$$

Ainsi le modèle qu'on va implémenter pour le véhicule sera donné par les équations suivantes:

$$\begin{cases} \dot{z}_1 = 0.012z_2 \\ \dot{z}_2 = \frac{g(60z_2)u}{60} - kz_2 \end{cases} \text{ avec } -1 \leq z_1 \leq 1; -1 \leq z_2 \leq 1 \text{ et } -1 \leq u \leq 1$$

L'espace dans lequel on cherchera la solution de la HJB se trouve ainsi normalisé et centré autour de l'origine.

Dans le programme qu'on va implémenter on va dénoter la distance que le véhicule doit traverser par 'd_max' et la vitesse maximale par v_max, la position vérifie

Chapitre III Programmation dynamique et résultats de simulation

$-\frac{d_{max}}{2} \leq x_1 \leq \frac{d_{max}}{2}$. Si on applique le même changement de variable on obtient le modèle mis à l'échelle suivant :

$$\begin{cases} \dot{z}_1 = \frac{2v_{max}}{d_{max}} z_2 \\ \dot{z}_2 = \frac{g(v_{max}z_2)}{v_{max}} u - kz_2 \end{cases}$$

Avec : $-1 \leq z_1 \leq z_2$; $-1 \leq z_2 \leq 1$ et $-1 \leq u \leq 1$.

Et la matrice de transformation de coordonnées est : $\begin{bmatrix} z_1 \\ z_2 \end{bmatrix} = \begin{bmatrix} \frac{1}{2} & 0 \\ 0 & \frac{1}{v_{max}} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$

Pour valider le programme implémenté, la partie qui calcul la commande optimale de systèmes continus a été utilisée pour calculer la commande d'un système de premier ordre et le résultat a été comparé avec la solution théorique.

Le système en question est défini comme suit :

$$\dot{x}(t) = x(t) + u(t); x(0) = 1$$

La commande doit minimiser le critère suivant :

$$J(x, u) = \frac{1}{4}x(t_f) + \frac{1}{4} \int_0^{t_f} u^2(t) dt$$

L'état initial $x(0) = 1$ et le temps final est fixé à $t_f = 10sec$. La solution théorique de ce problème est comme suit :

$$V(x, i) = \min_u J(x, u, t) = \frac{1}{2}K(t)x(t)$$

Avec $K(t) = \frac{e^{t_f-t}}{e^{t_f-t} + e^{-t_f+t}}$; la commande par retour d'état est donnée par :

$$u(t) = -2K(t)x(t)$$

La figure suivante donne l'évolution de l'état et la commande par retour d'état obtenue.

On remarque que plus on s'approche de t_f , plus le cumule des erreurs de la méthode d'intégration est plus grand et la méthode nécessite un maillage plus serré (un plus grand nombre de points) ce qui nécessite beaucoup plus de calculs et alourdit le programme.

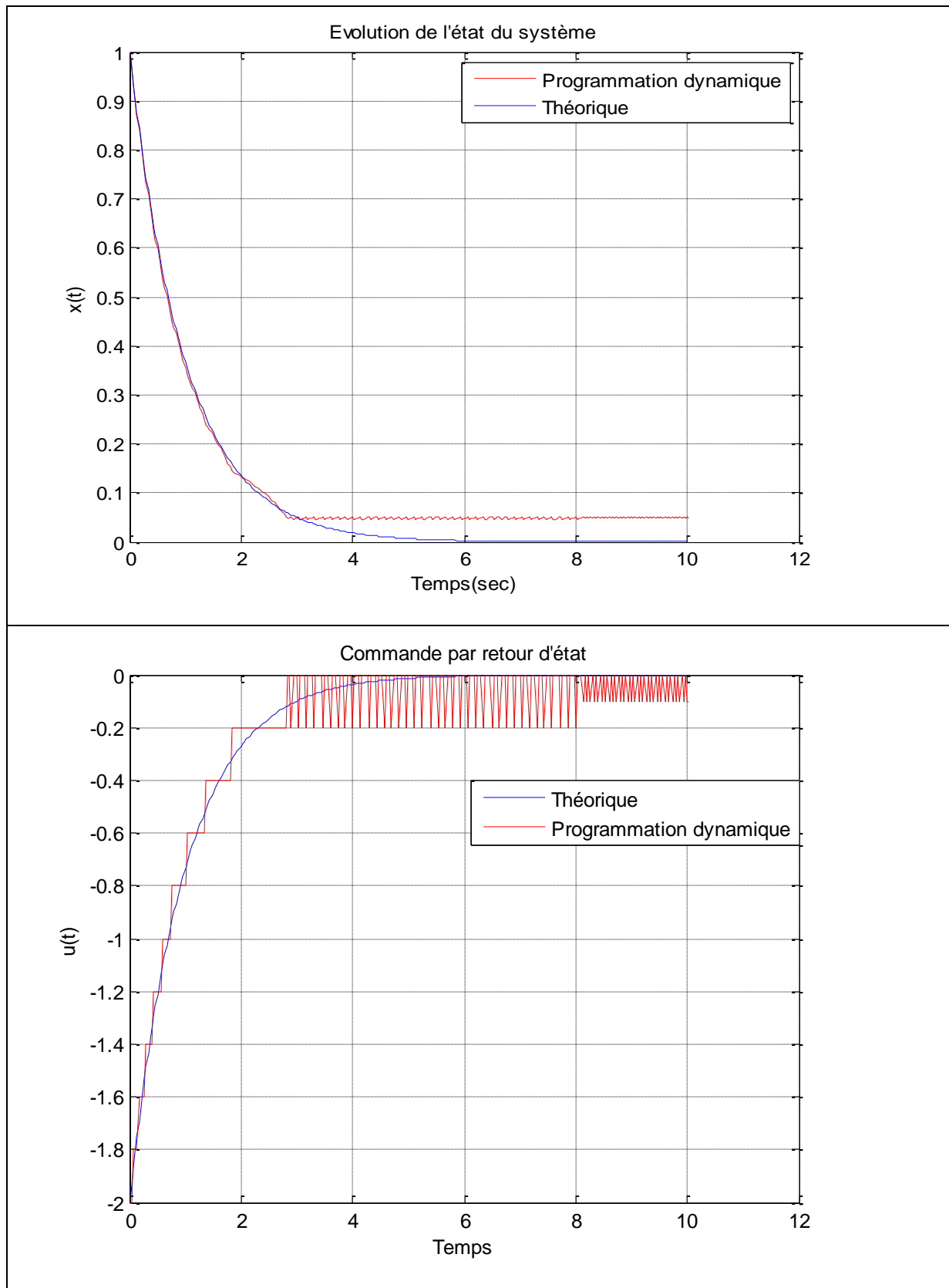
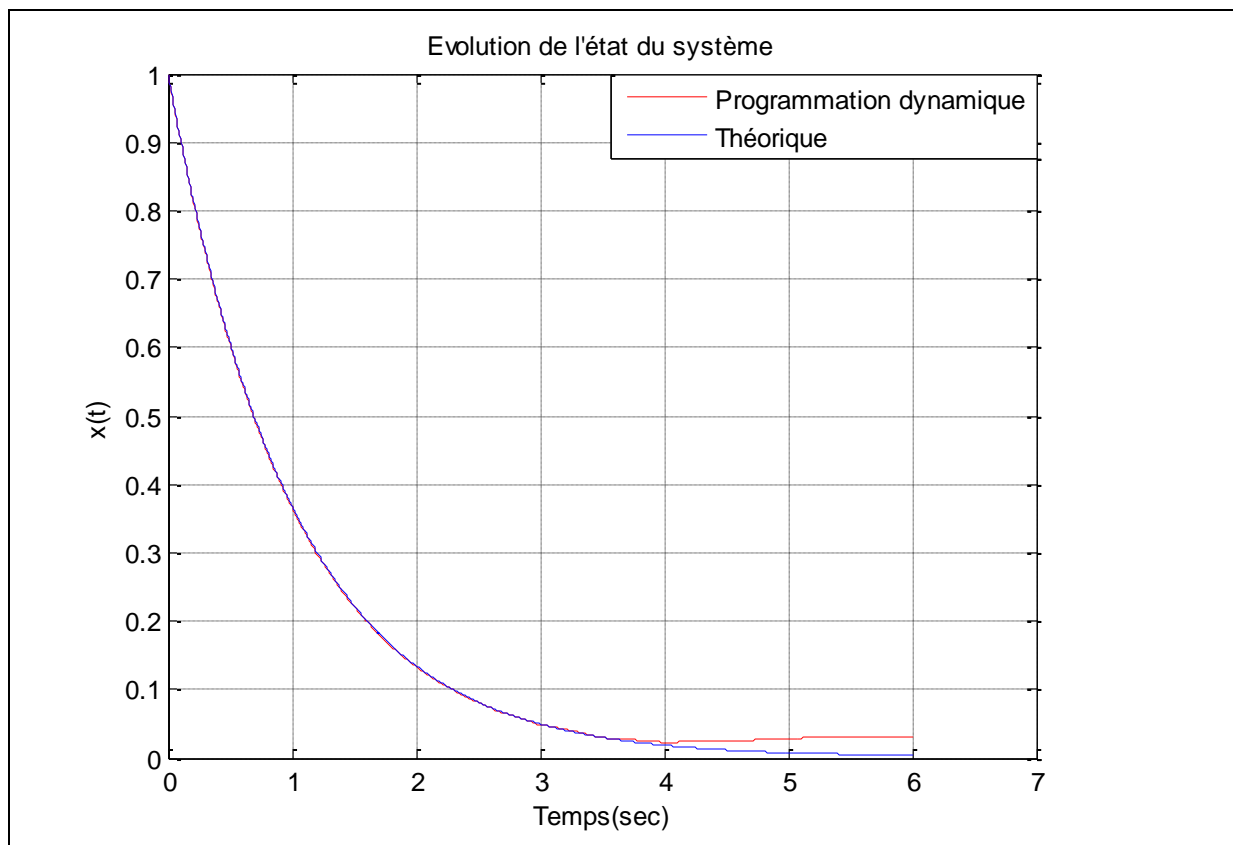


Figure (3.7) Evolution de l'état de système en boucle fermée et la commande par retour d'état obtenu par programmation dynamique de l'exemple du système continu. $\delta = 0.1$

Chapitre III Programmation dynamique et résultats de simulation

L'examen de la figure (3.7) montre que pour t proche de t_f (les dernières itérations de l'algorithme) le cumul des erreurs due à l'intégration devient important, ce qui se présente sous forme d'oscillation au niveau de la commande et d'offset au niveau de la fonction de valeur. Si le pas de maillage est plus petit (maillage serré) cette erreur va décroître. Mais ceci engendre une explosion au niveau de la dimension du maillage, ce qui nécessite un temps de calcul beaucoup plus important.

Le résultat obtenu pour le même système pour $\delta = 0.02$ est donné par la figure suivante :



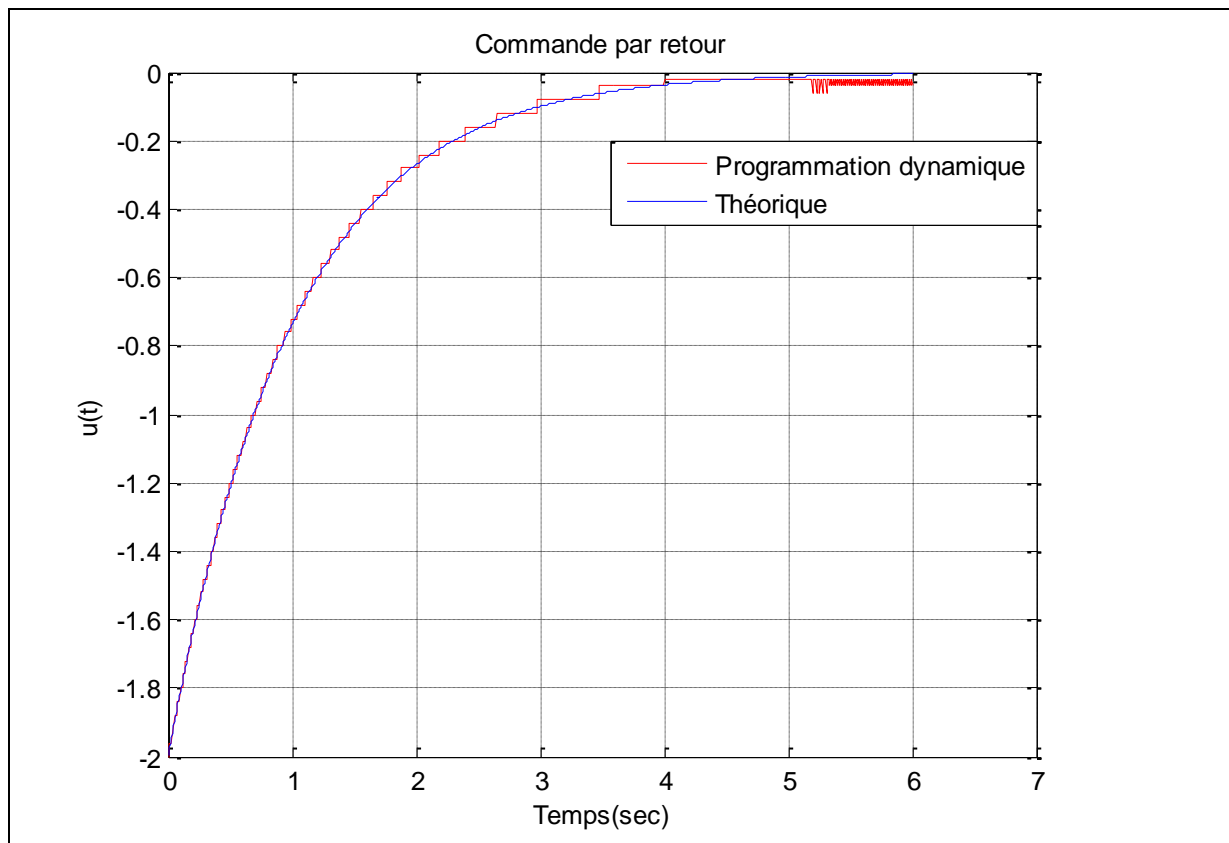


Figure (3.8): Evolution de l'état du système continu du premier ordre ainsi que la commande par retour d'état pour $\delta = 0.02$

Pour la commande optimale du véhicule, les caractéristiques du rendement du moteur pour chaque vitesse engagée sont prises comme sur la figure (3.9). Ce rendement doit être maximal durant tout le trajet, le contrôleur doit changer la vitesse engagée selon la vitesse de déplacement du véhicule et selon la caractéristique du rendement. Ceci doit être alors formulé dans l'expression de l'indice de performance. Pour cela nous avons inclus dans l'expression du lagrangien $l(x, q, t)$ la quantité $\frac{1}{g(x_2)+0.001}$. Cette quantité est minimisée si le rendement est maximal.

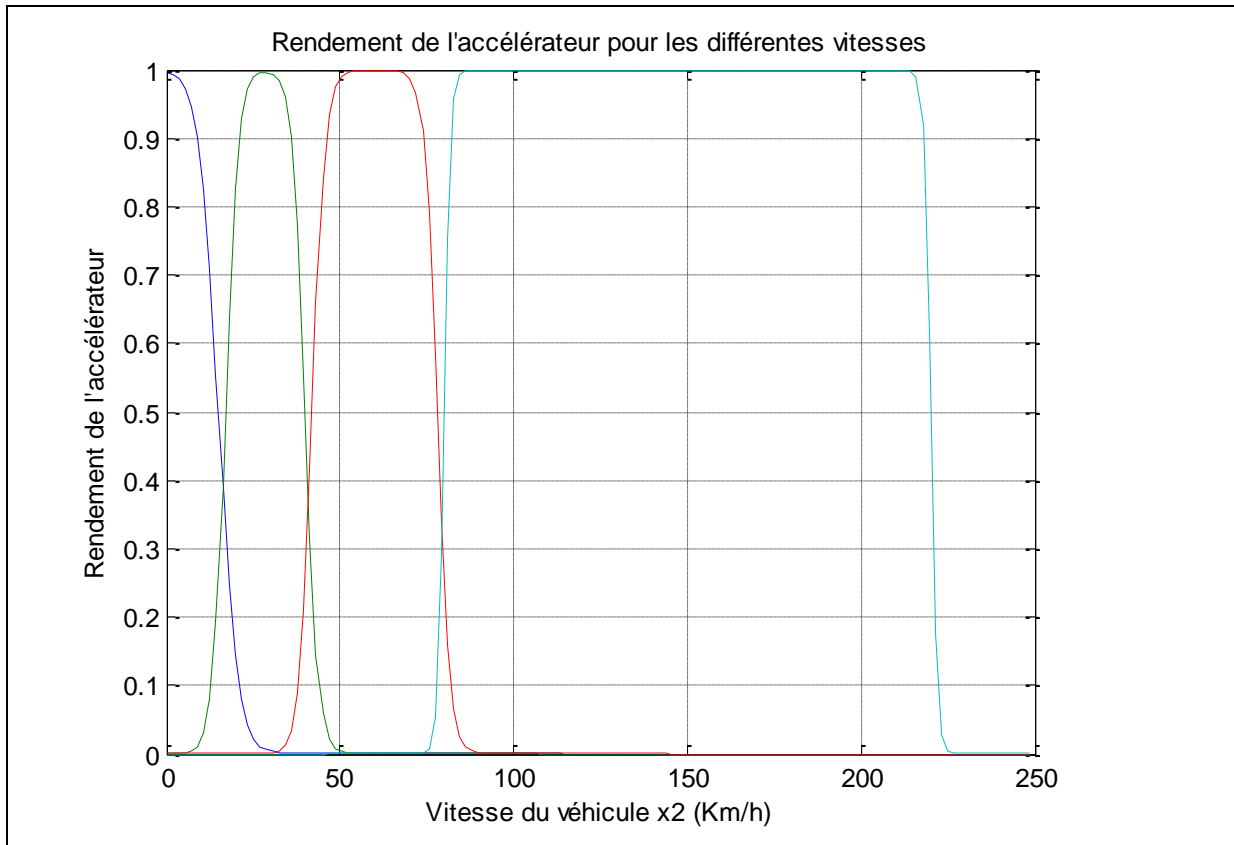


Figure (3.9) : Caractéristiques du rendement de l'accélérateur pour chaque vitesse.

Le véhicule doit traverser la trajectoire avec la vitesse la plus élevée possible. Ainsi la commande de l'accélérateur doit être élevée. Pour garantir ceci, nous avons inclus l'expression $(u - u_{max})^2$ dans l'indice de $l(x, q, t)$. La fonction lagrangienne est donnée par l'expression suivante :

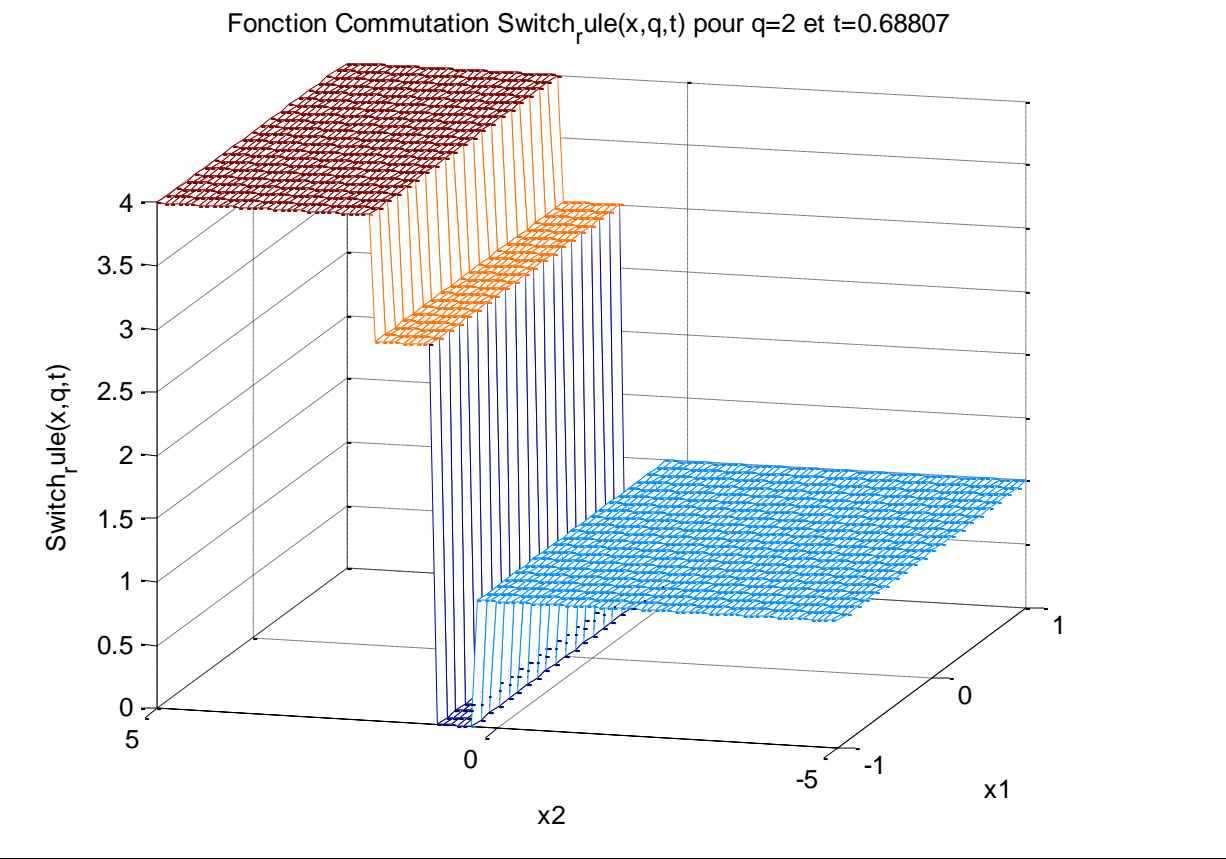
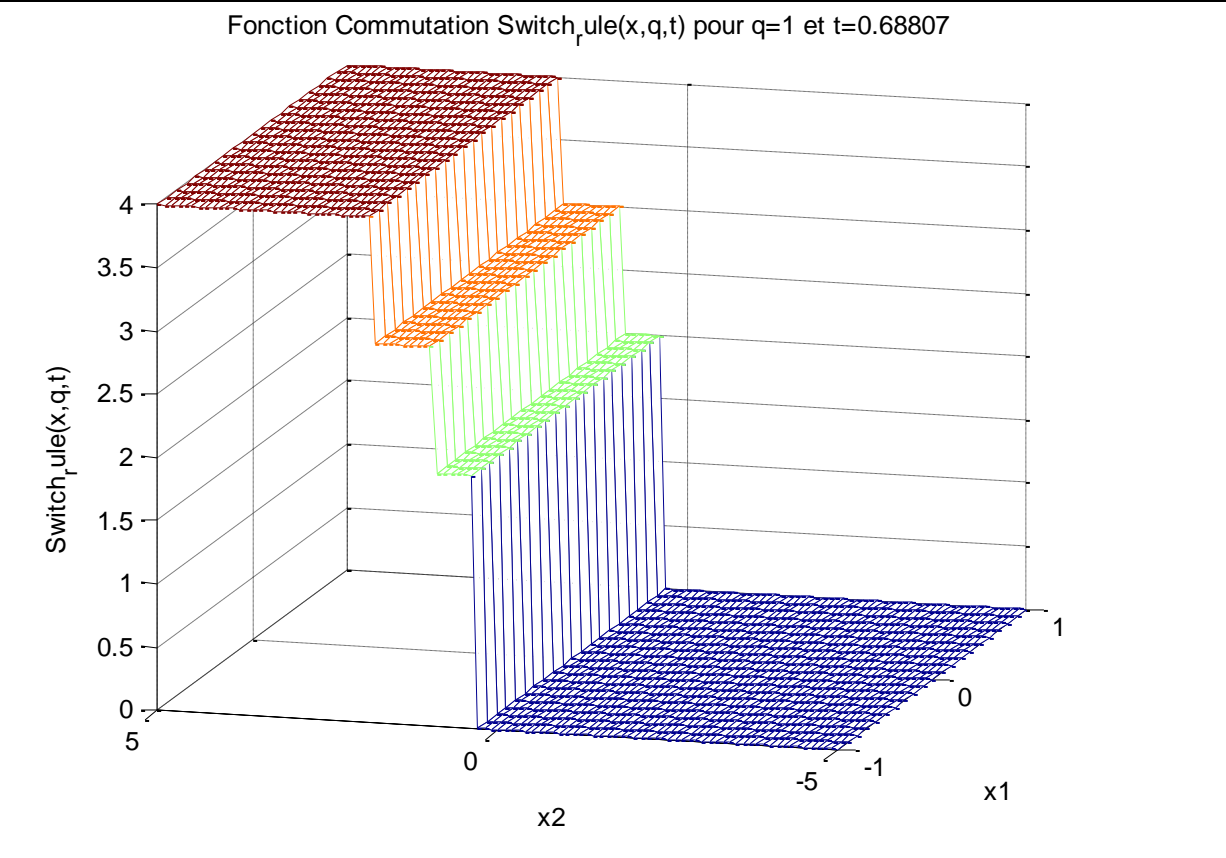
$$l(x, q, t) = 10^{-2}(600(u - u_{max})^2 + \frac{5}{g(x_1) + 0.001} + (x_1 - x_{1f})^2)$$

La fonction du coût terminal est choisie pour avoir $x_1(t_f) = x_{1f}$, son expression est $\psi(x, q, t) = (x_1(t_f) - x_{1f})^2 + x_2^2 + \frac{(q-1)^2}{100}$. Le coût des transitions discrètes est donné par la matrice suivante :

$$\kappa = \frac{1}{20} \begin{bmatrix} 0 & 0.4 & 0.6 & 0.7 \\ 0.4 & 0 & 0.4 & 0.6 \\ 0.6 & 0.4 & 0 & 0.4 \\ 0.7 & 0.6 & 0.4 & 0 \end{bmatrix}$$

Chapitre III Programmation dynamique et résultats de simulation

La loi de commutation $r_o(x, q, t)$ définie par l'équation (3.28), ne dépend que de x_2 , pour q et t donnée elle est indépendante de la position du véhicule. Cette fonction est égale à zéro si l'état actuel n'est pas un état de transition, et elle est égale à la valeur de l'état de destination si l'état en cours est un état de transition. Les figures suivantes donnent la forme de cette fonction pour les quatre états discrets et pour un t donnée.



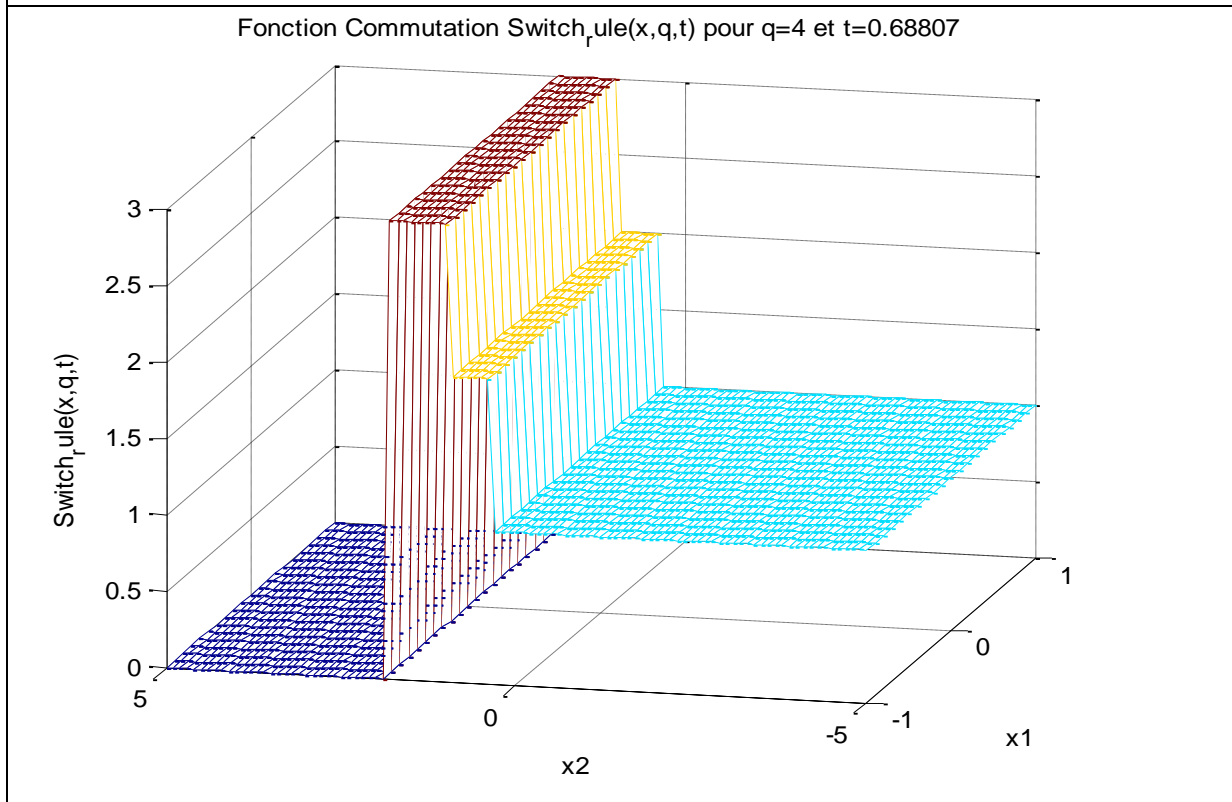
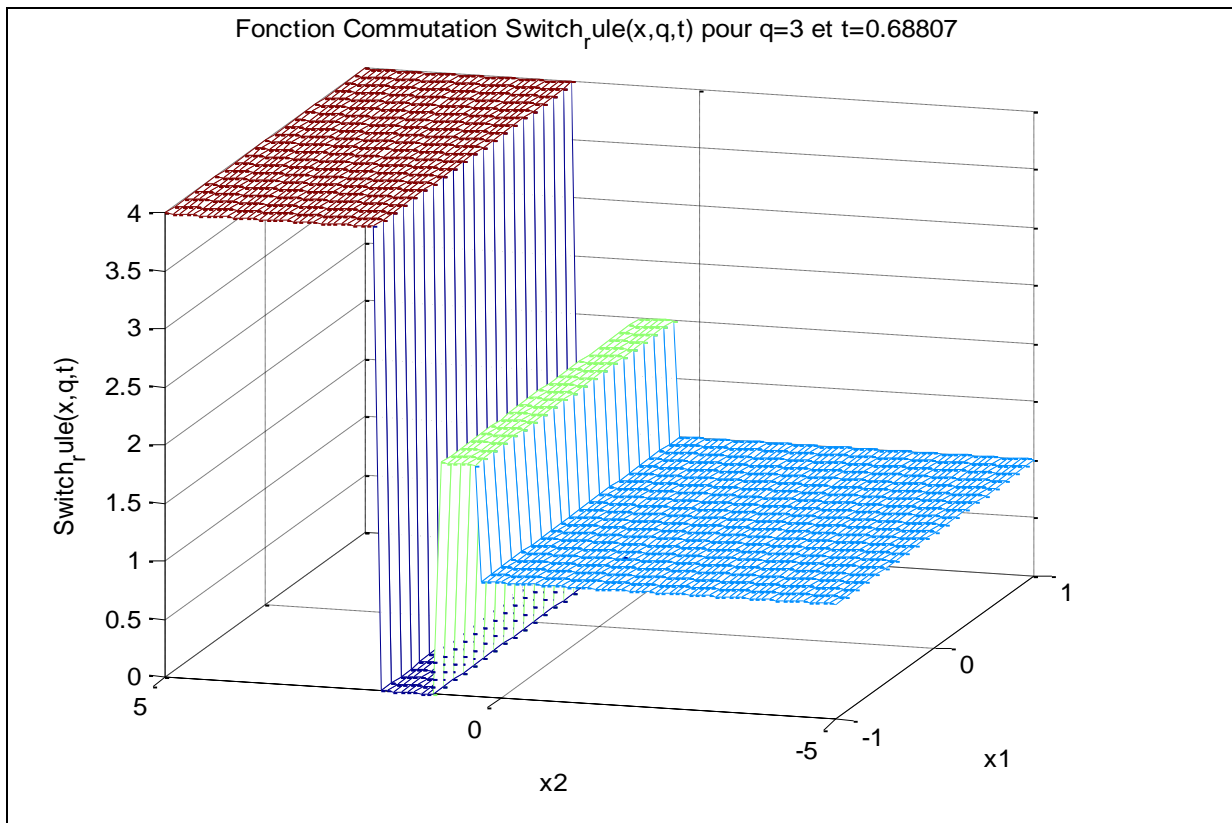


Figure (3.10) : Loi de commutation obtenue avec un maillage serré ($\delta = 0.1$).

Les autres paramètres sont fixés comme suit : $d_{max}=10000$; $v_{max}=12$; $t_0=0$;
 $q_{max}=4$; $t_f=5$; $d_{rond}=0.1$; $\lambda=1$; $x1_{min}=-1$;
 $x1_{max}=1$; $x2_{min}=-5$; $x2_{max}=5$; $u_{min}=-1$; $u_{max}=1$;

Chapitre III Programmation dynamique et résultats de simulation

Les deux figures suivantes représentent les résultats de simulation pour une distance traversée de 10Km. La première figure contient la commande $u(t)$, l'état discret $q(t)$, la position $x_1(t)$ et la vitesse de véhicule $x_2(t)$. On remarque que l'état discret (position du levier de vitesse) change en fonction de la vitesse de véhicule pour permettre un rendement maximal.

Le coût est, comme prévue, une fonction non croissante (sauf pour des points isolés).

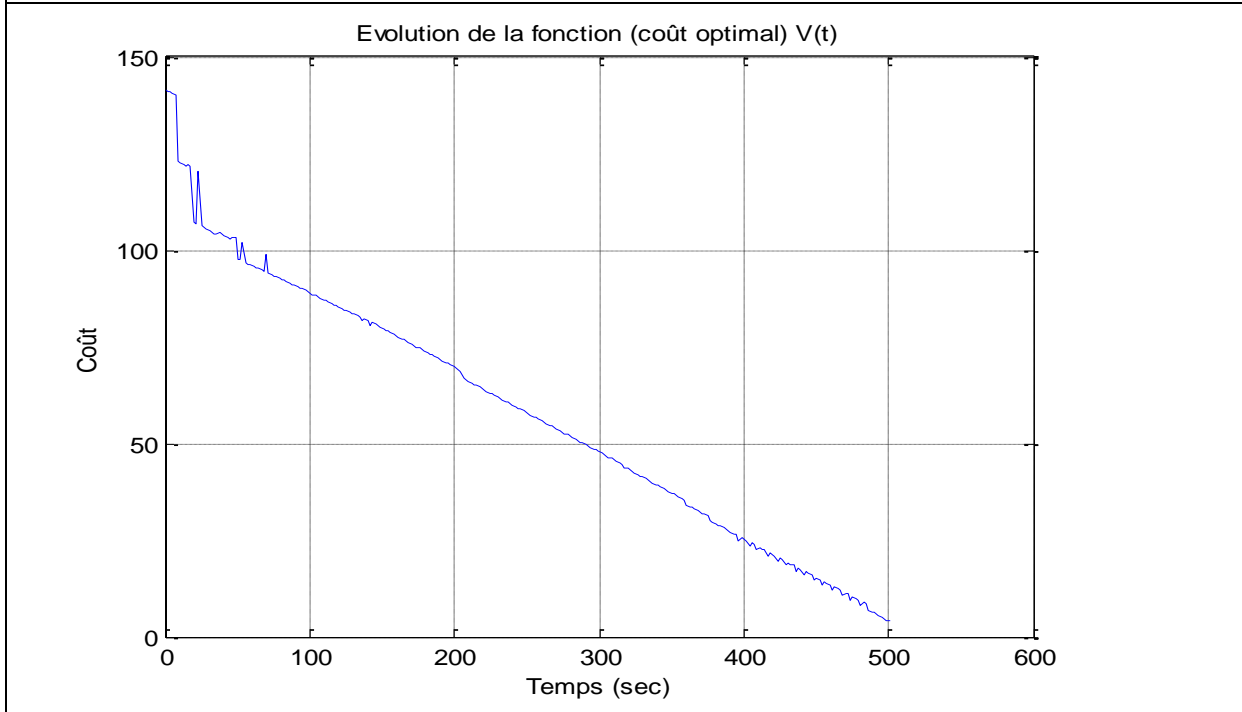
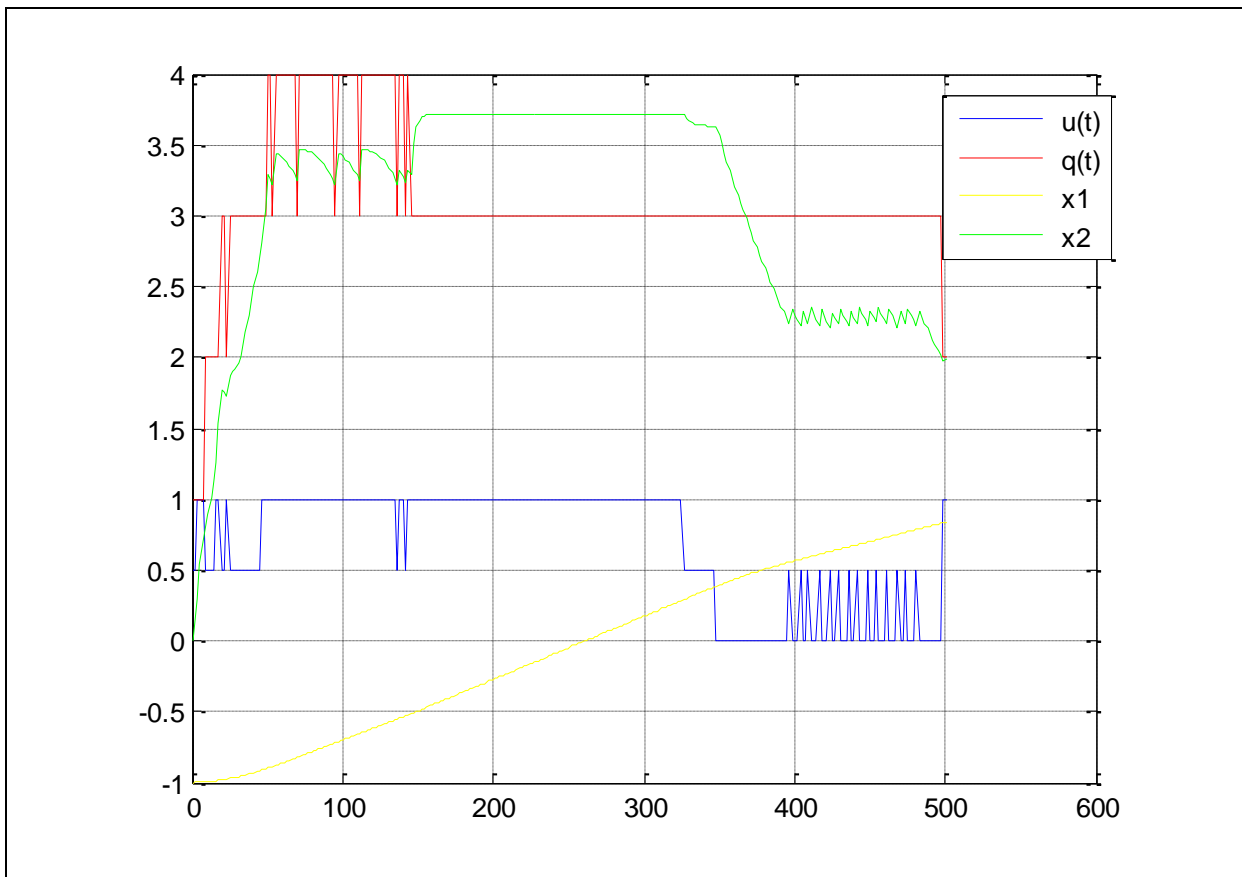


Figure (3.11) Variations des états continus $x(t)$, de l'état discret $q(t)$, de la commande $u(t)$ et de la fonction coût en fonction du temps.

Résultat obtenu pour $d_{max}=10000$; $v_{max}=60$; $v_{max}=v_{max}/10$; $t_0=0$;
 $q_{max}=4$; $t_f=500$; $d_{rond}=0.5$; $\lambda=1$; $x_1_{min}=-1$; $x_1_{max}=1$; $x_2_{min}=-10$;
 $x_2_{max}=10$; $u_{min}=-1$; $u_{max}=1$;
 $L=1e-2*(20*(u-u_{max})^2+5/(g+0.001)+(x(1)-0.8)^2)$;

Conclusion

D'après les résultats précédents on conclut que le choix de la commande optimale basée sur la programmation dynamique permet d'obtenir le control optimal et les instants de commutation du système.

La méthode de la commande optimale basée sur la programmation dynamique consiste à calculer tous les coûts existants du système, ensuite elle choisit une décision basée sur des critères d'optimisation.

Pour que le système soit efficace, la méthode doit limiter les transitions discrètes et les interdire complètement lors de l'efficacité maximale.

L'utilisation d'une méthode d'approximation permet de calculer et de choisir le coût optimal du système afin de tirer la commande, pour que le control soit très efficace il faut travailler sur la diminution et la négligence de l'erreur d'approximation de la solution des *SQVIs*.

Conclusion générale

Récemment les systèmes dynamiques hybrides ont fait l'objet de beaucoup d'attention, plusieurs méthodes et outils d'aide à la modélisation et à l'analyse de ces systèmes ont été développés. Les systèmes dynamiques hybrides ont des applications dans plusieurs disciplines comme, la robotique, l'électronique...etc. Ces systèmes peuvent être modélisés par des automates hybrides qui font intervenir deux types de dynamiques différentes, des dynamiques continues et des dynamiques discrètes qui interagissent entre elles. Les dynamiques continues sont modélisées par des équations différentielles par contre les dynamiques discrètes par un automate fini [3].

L'interaction des dynamiques continues et discrètes des SDH rend la conception d'un contrôleur efficace très délicate. Ainsi, le choix de la méthode de conception de la commande doit prendre en compte toutes les dynamiques existantes, afin de satisfaire les exigences et les contraintes fixées. Les avantages que présentent les aspects de la commande optimale ont fait d'elle un candidat idéal pour la synthèse de la contrôlabilité des SDH. Dans ce travail nous avons utilisé la programmation dynamique car elle est bien adaptée pour la dynamique non linéaire de notre système. Son principe consiste à choisir un contrôle optimal dans un domaine fini et discrétisé basé sur l'optimisation.

Les équations de HJB constituent un système d'inégalités quasis variationnelles qu'il faut résoudre par une méthode numérique, une fois les équations de HJB résolues, c'est-à-dire que la fonction de valeur est calculée, on peut tirer le contrôle optimal à partir de la solution trouvée.

Dans ce travail les résultats obtenus sont satisfaisants mais ils dépendent du pas de discrétisation. Pour obtenir des résultats précis, il faut choisir un pas très petit, ceci conduit à des calculs prohibitifs. Nous avons utilisé un algorithme pour l'approximation de la solution optimale à commutation, car pour l'instant, il n'existe pas vraiment une méthode de résolution des *SQVIs*. Afin d'obtenir une commande optimale efficace, il faut rendre l'erreur d'approximation de la solution très petite voir même négligeable.

Bibliographie

- [1] Jean-Louis FERRIER, Jean-Louis BOIMOND, « Systèmes dynamiques à évènement discret : du modèle à la commande », Notes de cours - Université d'Angers, juin 2004
- [2] Ludivine FRAN CART, Pierre THIROUIN, Elodie Chanthery, « les systèmes hybrides », Rapport de TIPE - 2005/2006
- [3] Fatima El Guezar « Modélisation et Simulation des Systèmes Dynamiques Hybrides Affines par Morceaux » thèse Doctorat université de Toulouse, 2009
- [4] Sudkamp, « Automates finis », publication 1997
- [5] François Laroussinie « Automates temporisés et hybrides » Notes de cours de l'école d'été temps réel 2003.
- [6] John Lygeros, 'Lecture Notes on Hybrid Systems', Lecture notes, Department of Electrical and Computer Engineering, University of Patras Rio, Patras, Greece, 2004.
- [7] Monika KUROVSZKY, « Etude des systèmes dynamiques hybrides par représentation d'état discrète et automate hybride », Thèse de Doctorat Université Joseph Fourier, GRENOBLE 1, 2002
- [8] Richard C. Dorf, "Optimal Control Systems", éditeur de la série University of California, Davis, 2003
- [9] Emmanuel Trélat, « Contrôle optimal », Notes de cours Master de Mathématiques, Université d'Orléans 2007/2008
- [10] Emmanuel Trélat, « Contrôle optimal : théorie et applications », Notes de cours, Université Pierre et Marie Curie (Paris 6) et Institut Universitaire de France ; 2011
- [11] Dimitri. P. Bertsekas, "Dynamic programming and optimal control" volume 1,2,3rd edition, Athena Scientific, 2005.
- [12] Donald E. Kirk "Optimal control theory : An introduction" Dover Publication Inc. 1998.
- [13] Martino Bardi, Italo Capuzzo-Dolcetta "Optimal control and viscosity solutions of Hamilton-Jacobi-Bellman equations" Birkhäuser, 1997.
- [14] Pierre Riedinger, « Contribution à la commande des systèmes complexes », thèse de doctorat, Université de Nancy, Institut National Polytechnique de Lorraine, juin 2010
- [15] Harold J. Kushner "Numerical methods for controlled stochastic delay systems" Birkhäuser 2000.

[16] Assad Al Alam, “Optimally Fuel Efficient Speed Adaptation”, Master’s thesis at automatic control, ACS Department, Royal Institute of technology, Stockholm, Sweden, 2007.

[17] Aimad MOKHTARI “Diagnostic des systèmes hybrides : développement d’une méthode associant la détection par classification et la simulation dynamique” thèse de doctorat, Université de Toulouse Institut Nationales des Sciences Appliquées de Toulouse, octobre 2007

[18] Laurentiu Hetel , « Stabilité et commande robuste des systèmes à commutation » thèse de doctorat, Université de Nancy, Institut National Polytechnique de Lorraine, novembre 2007