

ⵔⵉⵔⵉⵏⵉ ⵏ ⵓⵏⵉⵔⵉⵎⵉⵏⵉⵏⵉ ⵏ ⵓⵏⵉⵔⵉⵎⵉⵏⵉⵏⵉ ⵏ ⵓⵏⵉⵔⵉⵎⵉⵏⵉⵏⵉ

République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université Akli Mohand Oulhadj de Bouira
Faculté des Sciences et des Sciences Appliquées
Département de Mathématique



MEMOIRE DE MASTER en Mathématique

Spécialité : Recherche Opérationnelle

Thème

Méthodes numériques en contrôle optimal

Présenté par : BENSALÉM LOKMARI

Présenté par : BENALIA MOUADH ISSAME EDDINE

Devant le jury composé de :

M. Hamid Karim
Mme Ouidja Daya
M. Akkouche A

Président MAA, UAMO BOUIRA
Encadreur MAA, UAMO BOUIRA
Examineur MCB, UAMO BOUIRA

16 septembre 2022

REMERCIEMENTS

Au terme de ce travail nous tenons tout d'abord à remercier : Dieu, de nous avoir donné le courage, la force et la volonté pour achever ce modeste travail et de nous avoir aidée tout au long de nos années d'étude.

Nous tenions également à adresser toute notre gratitude à notre encadreur, Mme. Daya OUIDJA pour avoir orientée et enrichi notre travail.

Nous la remercions pour sa disponibilité, sa compréhension et sa patience, ses précieux conseils ainsi que son souci du détail, qui ont abouti à la réalisation de ce mémoire.

Nous remercions également les membres du jury pour avoir accepté d'examiner notre travail et de l'enrichir par leurs propositions.

Nous remercions sincèrement toutes les personnes qui, à un moment donné, ont cru en nous, nos familles, nos amis ainsi que les personnes qui nous ont soutenu de près ou de loin au cours de la réalisation de ce mémoire.

Dédicaces

A tous ceux qui un jour ont cru en moi.

à ceux qui m'ont encouragé , à ceux qui m'ont soutenu, à tous ceux qui m'ont conseillé , à ceux qui m'ont écouté , à mes professeurs, à mes camarades, à mes amis, à ma famille, à mes parents, ce modeste travail vous est dédié.

Je tiens également à adresser une dédicace toute particulière à mon père Benalia Ali et ma mère cheurfi hamida et mon ami Habib Allah Benaroudj . et pour terminer à Bensalem Lokmari ce fut un réel plaisir de travailler avec toi.

Benalia Mouadh Issame Eddine

Dédicaces

Mes dédicaces vont particulièrement pour mes chers parents, mon frère, mes sœurs ,ma famille ,mes amis ,tous ceux qui ont contribué de près ou de loin pour la réalisation de ce travail et tous ce qui m'ont accompagné et soutenu.

Bensalem Lokmari

Table des matières

Remerciements	1
Introduction	3
1 Généralités sur le contrôle optimal	4
1.1 Introduction à la commande optimal	4
1.2 Objet de la commande	5
1.3 Condition initiale du système	6
1.4 Le but de la commande	6
1.5 Class des commandes admissibles	6
1.5.1 Commande	6
1.5.2 commande Bang-bang	6
1.6 Critère de qualité	6
1.6.1 Temps optimal	7
1.6.2 Coût optimal	7
1.6.3 Problème de Mayer-Lagrange	8
1.7 Ensemble d'accessibilité	8
1.8 Propriétés essentielles du problème de contrôle optimal	8
1.8.1 Position du problème	8
1.9 Controlabilité	9
1.9.1 Controlabilité des systèmes linéaires autonomes	9
1.9.2 Critère de Kalman	10
1.9.3 Controlabilité des systèmes linéaires non-autonomes	11
1.10 Controlabilité des systèmes non-linéaires	11
1.10.1 Définitions et préliminaires	12
1.11 Stabilisation	14
1.12 Observabilité	15
1.12.1 Définition et critères d'observabilité	16
2 Résolution d'un problème de contrôle optimal	20
2.1 Introduction	20
2.2 Méthodes indirectes	20
2.2.1 Principe du Maximum de Pontryagin	20
2.2.2 Méthode de tir simple :	22

2.2.3	Méthode de tir multiple :	23
2.2.4	Rappels sur les méthodes de Newton	25
2.3	Méthodes directes	26
2.3.1	Présentation - formulation mathématique	26
2.3.2	Discrétisation totale "Tir direct" :	26
2.4	les caractéristiques des méthodes directes et indirectes.	28
2.4.1	Méthode directe :	28
2.4.2	Méthode indirecte :	28
2.5	Conclusion	34
3	Temps minimum d'une masse avec limite de force	35
3.1	Rappels sur les lois de la physique(Newton)	36
3.1.1	La première loi de Newton	36
3.1.2	La deuxième loi de Newton	36
3.2	Contrôlabilité	37
3.3	Résolution théorique	37
3.4	Application de la methode tir simple (Methode indirect) :	41
3.5	Résolution à l'aide du logiciel MATLAB :	43
3.6	Méthode indirecte	43
3.6.1	fonction tirsimple	43
3.7	Application de la discrétisation totale (Méthode directe) :	48
3.7.1	Résolution à l'aide du logiciel MATLAB :	48
	Conclusion	52
	Annexe	52
	A Méthode d'Euler explicite	53
	B Méthode de Runge Kutta	55
	C Théorie des points conjugués	56
	D La programmation quadratique séquentielle (SQP)	58
	E Conditions de Lagrange	59
	F Méthodes de pénalité	60
	G Types d'équations différentielles	61
G.0.1	Équation différentielle à variables séparées	61
G.0.2	équation différentielle linéaire d'ordre 1	62
	H méthode de Galerkin	63

Introduction Générale

La théorie du contrôle est une branche interdisciplinaire de l'ingénierie et des mathématiques. La théorie du contrôle analyse les propriétés des systèmes commandés, c'est-à-dire les systèmes dynamiques sur lesquels on peut agir au moyen d'un contrôle [1]. L'objectif alors est de déterminer un contrôle qui permet de transférer le système d'un état initial à un état final. Cet objectif doit être réalisé en optimisant une certaine fonction coût, appelée critère, tout en satisfaisant un certain nombre de contraintes, ce qu'on appelle problème de contrôle optimal. La théorie du contrôle optimal regroupe un nombre de sujets et de domaines extrêmement vastes : économie, mécanique, médecine, chimie, robotique, aéronautique et biologie, . . .etc. En effet, de nos jours, les systèmes automatisés font complètement partie de notre quotidien ayant pour but d'améliorer notre qualité de vie et de faciliter certaines tâches : système de freinage ABS, assistance à la conduite, contrôle des flux routiers, photographie numérique, contrôle de procédés chimiques, chaînes industrielles de montage, systèmes médicaux automatisés, guidage aérospatiaux, . . .etc.

La théorie moderne du contrôle optimal a commencé dans les années cinquante, avec la formulation du principe du maximum de Pontryagin, qui généralise les équations d'Euler-Lagrange du calcul des variations. Les différentes approches proposées dans la littérature pour la résolution d'un problème de contrôle optimal peuvent être scindées en deux classes [2]. La première, dite approche directe, consiste à transformer le problème de contrôle optimal en un problème d'optimisation statique en faisant une discrétisation totale [3] qu'on peut résoudre ensuite par des méthodes d'optimisation déterministe ou stochastique [4].

La deuxième approche, dite approche indirecte, consiste à dériver d'abord les conditions d'optimalité en utilisant la programmation dynamique. Les conditions d'optimalité d'un problème de contrôle optimal sont obtenues soit par la programmation dynamique. La programmation dynamique basée sur le principe de Bellman [5] conduit à l'équation de Hamilton-Jacobi-Bellman. Cette équation, donnée par une équation aux dérivées partielles, est une condition suffisante d'optimalité.

Depuis longtemps, des chercheurs se sont intéressés à la résolution des problèmes de contrôle optimal en approchant la solution des conditions d'optimalité, et

plusieurs méthodes ont été développées pour obtenir la solution numérique de l'équation de Hamilton-Jacobi-Bellman ; telle que la méthode d'approximation de Galerkin [6], les méthodes de collocation [7], ainsi que la solution numérique des équations de Hamilton-Pontryagin, en utilisant la méthode de tir [8], la méthode de tir multiple [9] et les méthodes de collocation indirectes [10].

Ces dernières années, des méthodes semi-analytiques pour la résolution des équations différentielles ont été développées dans la littérature [11]. Ces méthodes ont été utilisées par des mathématiciens pour résoudre une très grande variété d'équations : linéaires, non linéaires, homogènes et non homogènes. Elles permettent de déterminer la solution exacte, si elle existe, tandis que dans l'autre cas, si la solution exacte n'existe pas, ces méthodes donnent une solution approchée avec une très grande précision. La solution est obtenue en utilisant des processus itératifs en choisissant une approximation initiale déterminée en considérant les conditions initiales et/ou les conditions aux limites du problème considéré. Parmi ces méthodes, on peut citer la méthode de décomposition d'Adomian [12], la méthode des itérations variationnelles [13], la méthode de perturbation d'homotopie [14]. Ces méthodes itératives sont aussi utilisées pour la résolution des problèmes de contrôle optimal décrits par les équations différentielles, en résolvant l'équation de HJB [15], les équations de Hamilton-Pontryagin [16]. La méthode de perturbation d'homotopie introduite par le mathématicien chinois Ji Huan He, en 1998, a été démontrée par plusieurs chercheurs qu'elle est fiable et efficace à des fins analytiques et numériques. Des études de comparaison rapportées dans la littérature révèlent la supériorité de la méthode et sa simplicité. Ceci est justifié par le nombre important des applications de cette méthode dans différents domaines.

Le reste du mémoire est structuré comme suit :

Le premier chapitre sera consacré à une introduction de contrôle optimal. Nous avons présenté les notions essentielles d'un programme de contrôle optimal. La contrôlabilité, la stabilité et l'observabilité, après quelques définitions de base d'un problème de contrôle optimal,

Le deuxième chapitre est consacré à la présentation de quelques méthodes numériques pour la résolution.

Dans le troisième chapitre, on présente une application de la méthode directe et l'indirecte pour la résolution des problèmes de contrôle optimal.

Chapitre 1

Généralités sur le contrôle optimal

Introduction

La conception de système de contrôle classique est généralement un processus d'essais et d'erreurs dans lequel diverses méthodes d'analyse sont utilisées de manière interactive pour déterminer les paramètres de conception d'un système "acceptable". Les performances acceptables sont généralement définies en termes de critères de domaine temporel et fréquentiel tels que comme temps de montée, temps de stabilisation, dépassement de crête, marge de gain et de phase et bande passante. Cependant, des critères de performance radicalement différents doivent être satisfaits par les systèmes complexes à entrées multiples et à sorties multiples nécessaires pour répondre aux exigences de la technologie moderne. Par exemple, la conception d'un système de contrôle d'attitude d'un engin spatial qui minimise la dépense de carburant ne peut pas être résolu par des méthodes classiques, Une approche nouvelle et directe de la synthèse de ces systèmes complexes, appelée contrôle optimal, a été rendue possible par le développement de l'ordinateur numérique. L'objectif de la théorie du contrôle optimal est de déterminer les signaux de contrôle qui amèneront un processus à satisfaire les contraintes physiques et en même temps à minimiser (ou maximiser) certains critères de performance. problème de contrôle », mais considérons la question de la formulation du problème.

1.1 Introduction à la commande optimal

La problématique générale d'un problème de contrôle optimal est donnée par un système d'équations différentielles dont le comportement dépend de variables extérieures dites variables de contrôle (ou commande). Pour un contrôle donné, on définit la trajectoire associée, ainsi qu'un critère à optimiser.

Nous considérons un problème dit de Bolza suivant ;

$$\text{Minimiser } J(u(t_f), t_f) = g^\circ(x(t_f), t_f) + \int_0^{t_f} f^\circ(u(t), x(t), t) dt, \quad (1.1)$$

Sous les contraintes :

$$\dot{x}(t) = f(u(t), x(t), t), \quad (1.2)$$

$$x(0) = x_0 \in M_0, \quad (1.3)$$

$$x(t_f) = x_f \in M_f, \quad (1.4)$$

$$u(t) \in U, t \in T = [0, t_f], \quad (1.5)$$

Où

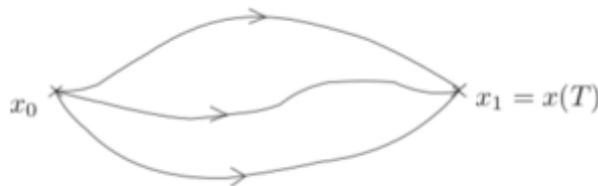
- $J(u(t_f), t_f)$ est le critère à optimiser (autrement appelé cout, critère de qualité, fonctionnelle), avec $f^\circ : U \times V \times T \rightarrow R$ est une fonction de classe C^1 , U ouvert de R^m , V ouvert de R^n , T est un intervalle temps. Cette fonction dépend de $x(\cdot)$ appelée équation d'état et de $u(\cdot)$ qui est la variable de contrôle sur le système.

$g' : V \times T \rightarrow R$ est une fonction continue. Elle dépend de t_f qui est le temps final (fixé ou libre) et de l'état final $x(t_f)$.

- $f : U \times V \times T \rightarrow R^n$ est une application de classe C^1 . L'équation (1.2) appelée état contrôlé du système, ou trajectoire contrôlée du système.

- x_0 est la position initiale du système, x_f est la position finale du système, et M_0 et M_f sont deux variétés de R^n .

- U est l'ensemble des applications mesurables, localement bornées sur T à valeurs dans l'ensemble non vide $\Omega \subset R^m$



1.2 Objet de la commande

On suppose que l'état x est solution d'une équation différentielle :

$$\frac{\partial x}{\partial t} = f(t, x(t), u(t)), \quad x(0) = x_0 \quad (1.6)$$

où $x(t) \in R^n$ est la position de l'objet à l'instant t , $x(t_0) \in R$ la position initiale du système, et $u(t)$ la commande

1.3 Condition initiale du système

Définition 1.1. La condition initiale du système, $x_0 = x(0)$ est un vecteur donné dans un plan de phase. $x(t)$ et $x(0)$ peuvent représenter physiquement la vitesse, la température,..etc.

1.4 Le but de la commande

Dans un problème de contrôle, le but de la commande consiste à ramener l'objet considéré de la position initial $x_0 = x(t_0)$, ($x_0 \in G_0$) à la position $x_1(t_1)$, ($x_1 \in G_1$), (où G_0 est l'ensemble de départ, G_1 est l'ensemble d'arrivée (accessibilité)).

1.5 Class des commandes admissibles

La classe des commandes admissibles U est constituée de fonctions mesurables $u(t)$:

$$U = \{u(t), \quad t \in T = [0, t_f]\}.$$

Chaque commande transfère l'objet du point de départ x_0 en un point de l'ensemble d'arrivée $G(t_f)$.

1.5.1 Commande

Dans beaucoup de problèmes de contrôle, on peut minorer et majorer les $u_j(t)$ par des constantes. Dans la suite, nous considérons ce type de problème avec $a_j \leq u_j \leq b_j$. Notons que l'on peut remplacer u_j par v_j on posant $u_j = \frac{1}{2}(a_j + b_j) + \frac{1}{2}(a_j - b_j)v_j$ et ainsi v_j est aussi intigrable et l'on a $-1 \leq v_j \leq 1$. Danc lorsque U est borné, il est toujours pratique de ce ramener à des commandes entre -1 et 1.

1.5.2 commande Bang-bang

On suppose que U est un polyèdre (cube) $[-1, 1]^m$ dans R^m . Un contrôle $u \in U$ est appelé contrôle Bang-Bang si pour chaque instant t et chaque indice $j = 1, \dots, m$ on a $|u_j(t)| = 1$. En d'autres termes, une commande Bang-Bang est une commande qui possède ou moins un switch.

1.6 Critère de qualité

L'objectif, lors de la formulation d'un problème de contrôle, est de fournir la motivation physique pour la sélection d'une mesure de qualité pour le système.

Le problème revient à définir une expression mathématique qui, lorsqu'elle est optimisée, indique que le système atteint un état désirable. Donc, choisir une mesure de qualité, est une traduction en termes mathématiques des exigences physiques du système. Le critère de qualité, appelé aussi coût ou fonction objectif, est généralement décrit par la formule

$$J(x, u) = g(t_f, x_f) + \int_0^{t_f} f_0(t, x, u) dt \quad (1.7)$$

Cette fonctionnelle comporte deux parties : $g(t_f, x_f)$ est le coût terminal, c'est une sorte de pénalité liée à la fin de l'évolution du système au temps final t_f ; il a son importance lorsque t_f est libre, si non il est constant. Le second terme intervenant dans la fonction objectif $\int_0^{t_f} f_0(t, x, u) dt$ dépend de l'état du système tout au long de la trajectoire de la solution, définie par les variables d'état. Cette trajectoire dépend aussi du temps t mais surtout des variables de contrôle u . C'est une fonction d'efficacité de chaque commande sur l'intervalle T . On peut classer les fonctions objectifs en deux critères physiques de performance :

1.6.1 Temps optimal

On parle d'un problème en temps optimal lorsque $f_0(t, x, u) = 1, g(t_f, x_f) = 0$ et le temps final t_f est libre dans l'expression de $\min \int_0^{t_f} 1 dt$.

1.6.2 Coût optimal

On parle d'un problème en coût optimal lorsque le temps final t_f est fixé dans l'expression

$$\min g(t_f, x_f) + \int_0^{t_f} f_0(t, x, u) dt \quad (1.8)$$

Évidemment, il existe des problèmes qui combine les deux critères physiques de qualité, et on parlera dans ce cas d'un problème de contrôle en temps et en coût optimal. Si dans l'expression de J , f_0 est proportionnelle à u^2 , on parle alors d'un coût quadratique.

Si u est un contrôle scalaire et f_0 est proportionnelle à u , on parle de problème de contrôle à coût d'approvisionnement. Lorsque les équations d'état $x' = f(x, u)$ ne dépendent pas explicitement de la variable t , on parle dans ce cas de problème autonome.

Si t est présent dans les équations d'état on parle de problème non-autonome.

1.6.3 Problème de Mayer-Lagrange

Le problème de Mayer-Lagrange est donné sous la forme d'un système

$$\dot{x}(t) = f(t, x(t), u(t)), x(0) = x_0, x(t_f) = x_f, u \in U, t \in [0, t_f], \quad (1.9)$$

l'objectif étant de minimiser le coût

$$J(t_f, u) = g(t_f, x_f) + \int_0^{t_f} f_0(t, x, u) dt. \quad (1.10)$$

Lorsque $g = 0$ dans l'expression de la fonctionnelle J , on parlera d'un problème de Lagrange, lorsque $f_0 = 0$, on parlera d'un problème de Mayer.

1.7 Ensemble d'accessibilité

Considérons le système contrôlé suivant :

$$\frac{dx}{dt} = \dot{x}(t) = A_x(t) + B_u(t), x(t_0) = x_0. \quad (1.11)$$

L'ensemble des points accessibles à partir de x_0 en un temps $t_f > 0$ est :

$$A(x_0, t_f) = \{x_1 \in R^n / \exists u \in C_{morceaux}^0([0, t_f], U),$$

$$\exists x : R \rightarrow R^n, C^1 \text{ morceaux } t, q \ x(0) = x_0,$$

$$\forall t \in [0, t_f] \ x'(t) = A_x(t) + B_u(t), x(t_f) = x_1.$$

Autrement dit $A(x_0, t_f)$ est l'ensemble des extrémités des solutions de (1.11) au temps t_f , lorsqu'on fait varier le contrôle u

1.8 Propriétés essentielles du problème de contrôle optimal

1.8.1 Position du problème

Dans la classe des commandes constantes par morceaux

$$U = \{u(t), t \in T = [t_0, t_f]\},$$

considérons le problème de contrôle optimal suivant :

$$\begin{cases} J(u(t)) = c'x(t_f) + \int_0^{t_f} f_0(x, u, t) dt \rightarrow \max_{u \in U}, & (1) \\ \dot{x} = \frac{dx(t)}{dt} = A_x(t) + b_u(t), x(0) = x_0, & (2) \\ H_x(t_f) = g, & (3) \\ |u(t)| \leq 1; t \in [0, t_f]. & (4) \end{cases}$$

- $J(u)$ est le critère de qualité (fonction objectif).
- $x(t) = (x_j, j \in J) \in R^n$, un n -vecteur représentant la position (l'état) du système à l'instant $t \in T$.
- $A(I, J)$ est une $n \times m$ -matrice qui caractérise le système.
- $b(J)$ est un n -vecteur donné.
- $H = H(I, J)$ C'est une $m \times n$ -matrice telle que le rang de $h = m \leq n$.
- g est un m -vecteur qui représente la sortie du signal à l'instant t_f .
- $u(t)$ est une commande.
- $C = c(J)$ est le n -vecteur des coûts.
- $I = \{1, \dots, m\}$ et $J = \{1, \dots, n\}$ sont des ensembles d'indices.
- x_0 la position initiale.

1.9 Controlabilité

Pour certains types de problème, avant leur résolution, on s'intéresse à l'existence de leurs solutions en utilisant les notions de commandabilité ou de contrôlabilité. Elle consiste à faire passer le système d'un état initial x_0 à un état final x_1 prescrit en un temps fini. Une fois le problème de contrôlabilité résolu, on peut vouloir passer de l'état initial à l'état final en minimisant un certain critère. On veut répondre à la question suivante : Etant donné le système (1.9), où peut-on aller en temps t_f en faisant varier le contrôle u . On rappelle tout d'abord la notion d'ensemble accessible.

1.9.1 Controlabilité des systèmes linéaires autonomes

Théorème 1.1. [26] *Un système linéaire autonome de R^n :*

$$\dot{x}(t) = A_x(t) + B_u(t),$$

est contrôlable en temps t_f si et seulement si le rang de la $n \times n$ matrice

$$K = [B, AB, A^2B, \dots, A^{n-1}B],$$

est égal à n .

L'essentiel de la preuve est contenu dans le lemme suivant :

Lemme 1.2. [26] *La matrice K est de rang n si et seulement si l'application linéaire*

$$\Phi' : L^\infty([0, t_f], R^m) \rightarrow R^n$$

$$u \rightarrow \int_0^{t_f} e^{(t_f-t)A} B_u(t) dt, \quad \text{est surjective.}$$

Remarque 1.1. La matrice K est appelée matrice de Kalman et la condition $\text{rang}(K) = n$ est appelée condition de Kalman, elle ne dépend pas de la condition initiale x_0 , ni de temps final t_f . Ceci signifie que si un système linéaire autonome est contrôlable en temps t_f depuis x_0 , alors il est contrôlable en tout temps depuis tout point. Ce théorème donne ainsi une condition nécessaire et suffisante de contrôlabilité dans le cas où les matrices A et B ne dépendent pas de la variable de temps t .

Exemple 1.1. Considérons le système dynamique linéaire autonome suivant :

$$\begin{cases} \dot{x}_1(t) = x_2(t), \\ \dot{x}_2(t) = u(t) \end{cases}$$

Avec $\dot{x}(t) = A_x(t) + B_u(t)$

où

$$A = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix}, B = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

Pour vérifier la contrôlabilité de ce système, il suffit de calculer le déterminant de la matrice de Kalman. Par conséquent, la matrice de Kalman K est donnée par :

$$K = [B, AB] = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$$

Le déterminant de K est égal à : $\det(K) = -1 \neq 0$ donc le $\text{rang}(K) = 2$, d'où le système est contrôlable.

1.9.2 Critère de Kalman

Soit $T > 0$ un horizon temporel fixé. On considère un système dynamique dont l'état $x(t) \in R^n$ pour tout $t \in [0, t_f]$ est régi par le système différentiel

$$\begin{cases} \dot{x} = Ax(t) + Bu(t) & t \in [0, t_f] \\ x(0) = x_0 \end{cases} \quad (1.2)$$

Où $A \in M_n(R)$, $B \in M_n m (R)$ et $u(.) : [0, t_f] \rightarrow R^m$ est une fonction de contrôle. Puisque la trajectoire x , solution de (1.2), dépend du contrôle u , nous la noterons souvent x_u . Il n'est pas toujours pertinent de chercher un contrôle u continu, on ne cherchera pas une trajectoire de classe C^1 ($[0, t_f]; R^m$). Un bon cadre fonctionnel pour la trajectoire est celui des fonctions absolument continues¹ sur $[0, t_f]$ [1]

Théorème 1.3. [23] (Critère de Kalman)

Le système linéaire autonome (1.2) est (exactement) contrôlable pour tout $t_f > 0$ et pour tout $x_0 \in \mathbb{R}^n$ si et seulement si la matrice Kal est de rang maximal, i.e. $rg(Kal) = n$.

On dit qu'une fonction $F : [0, t_f] \rightarrow \mathbb{R}^n$ est absolument continue sur $[0, t_f]$ et on écrit $F \in AC([0, t_f]; \mathbb{R}^n)$ s'il existe $f \in L^1([0, t_f]; \mathbb{R}^n)$ telle que

$$F(t) - F(0) = \int_0^t f(t)dt \quad \forall t \in [0, t_f]$$

Si une fonction F est absolument continue sur $[0, t_f]$, alors elle est continue sur $[0, t_f]$ et elle est dérivable presque partout, de dérivée égale à f .

Remarque 1.2. La condition de Kalman est indépendante de l'horizon temporel $t_f > 0$ et de la donnée initiale x_0 , ce qui implique en particulier que lorsqu'un système de contrôle linéaire autonome est contrôlable, on peut atteindre à partir d'une donnée initiale toute cible en un horizon temporel même très court. Attention toutefois : le prix à payer est que le contrôle peut prendre des valeurs arbitrairement grandes.

1.9.3 Controlabilité des systèmes linéaires non-autonomes

Théorème 1.4. [26] Un système linéaire non-autonome de \mathbb{R}^n :

$$\dot{x}(t) = A(t)x(t) + B(t)u(t),$$

est contrôlable en temps t_f si et seulement si la matrice

$$C = \int_0^{t_f} (F(t))^{-1} B(t) B'(t) (F'(t))^{-1} dt,$$

est inversible.

Remarque 1.3. La matrice C est appelée matrice de contrôlabilité. Elle ne dépend pas de la condition initiale x_0 , mais elle dépend du temps final t_f .

1.10 Controlabilité des systèmes non-linéaires

Se prononcer sur cette propriété reste jusqu'à présent une tâche très difficile. Pour étudier la commandabilité des systèmes non linéaires, on a tendance à utiliser

le systèmes linéarisé partant du fait que la commandabilité du système linéarisé implique celle du système non linéaire d'une manière locale [27]. La non commandabilité du système linéarisé n'implique pas forcément la non commandabilité du système non linéaire [27]. Ceci constitue l'inconvénient principal de cette méthode. Considérons un système de contrôle non linéaire suivant :

$$\begin{cases} \dot{x}(t) = f(t, x(t), u(t)) & t \in [0, t_f] \\ x(0) = x_0, \end{cases} \quad (1.12)$$

1.10.1 Définitions et préliminaires

Etant donné un point $x_1 \in R^n$, on se propose de trouver un temps t_f et un contrôle u sur I tel que la trajectoire x_u associée à u , vérifie :

$$x_u(0) = x_0, x_u(t_f) = x_1. \quad (1.13)$$

Ceci conduit à la définition suivante :

Définition 1.2. Soit $t_f > 0$, L'application entrée-sortie en un temps t_f du système de contrôle non-linéaire initialisé à x_0 est l'application :

$$\begin{aligned} E_{x_0, t_f} : L^\infty([0, t_f], R) &\rightarrow R^n \\ u &\rightarrow x_u(t_f) = x_1, \end{aligned} \quad (1.14)$$

Où $x_u(\cdot)$ est la solution du système

$$\begin{cases} \dot{x}_u(t) = Ax_u(t) + Bu(t), \\ x_u(0) = x_0. \end{cases} \quad (1.15)$$

U est l'ensemble des contrôles u tels que la trajectoire associée est bien définie sur $[0, t_f]$.

Lemme 1.5. L'application E_{x_0, t_f} est différentiable au sens de Fréchet. Et sa différentielle au sens de Fréchet à l'ordre un est :

$$E_{x_0, t_f}(u + \delta u)(t) = E_{x_0, t_f}(u) + dE_{x_0, t_f}(u) \cdot \delta u + o(\delta u L^\infty). \quad (1.16)$$

Par définition :

$$\begin{aligned} \dot{x}_u(t) &= Ax_u(t) + B(u(t)); \quad x_u(0) = x_0, \\ x_u + \delta u(t) &= Ax_u + \delta u(t) + B(u + \delta u)(t), \quad x_{u+\delta u}(0) = x_0. \end{aligned}$$

On pose

$$\delta x(t) = x_{u+\delta u}(t) - x_u(t),$$

$$\delta\dot{x} = A\delta x + B(u + \delta u) - B(u), \quad \delta x(0) = 0,$$

avec

$$B(u + \delta u) = B(u) + dB(u).\delta u + o(\delta u).$$

ainsi on aura le système dit linéarisé le long de $u(\cdot)$ à l'ordre un suivant :

$$\begin{cases} \delta'x(t) = A\delta x(t) + dB(u(t))\delta u(t).\delta u(t), \\ \delta x(0) = 0. \end{cases} \quad (1.17)$$

Or à l'ordre un on a :

$$\delta x(t) = \int_0^t e^{(t-s)A} dB(u(s))\delta u(s) ds \quad (1.18)$$

D'où le résultat suivant :

$$dE_{x_0, t_f} = \int_0^{t_f} dB(u(s))\delta u(s) ds = \delta x(t_f) \quad (1.19)$$

Résultats de contrôlabilité

Définition 1.3. L'ensemble accessible en temps t_f pour le système (1.12), noté $A(t_f)$, est l'ensemble des extrémités au temps t_f des solutions du système partant de 0. Autrement dit, c'est l'image de l'application entrée/sortie en temps t_f

Définition 1.4. Le système (1.12) est dit contrôlable (en temps quelconque) depuis x_0 si :

$$R^n = \bigcup_{t_f \geq 0} A(t_f).$$

Par des arguments du type des fonctions implicites, l'étude de la contrôlabilité du système linéarisé (qui est plus simple), permet de déduire des résultats de contrôlabilité locale du système de départ, sur ce sujet le lecteur peut consulter [[22],[28]].

Théorème 1.6. [1] *Le système linéarisé (1.17) est contrôlable en temps t_f si et seulement si la matrice :*

$$\int_0^{t_f} e^{-tA} dB(u(t)) dB(u(t))^T e^{-tA^T} dt. \quad (1.20)$$

est inversible, A_T est la transposée de la matrice A .

Remarque 1.4. La condition de contrôlabilité dépend de t_f , mais ne dépend pas du point initial x_0 . Autrement dit, si le système considéré est contrôlable depuis x_0 , en temps t_f , alors il est contrôlable depuis tout point.

1.11 Stabilisation

Un contrôle (ou une commande) en boucle ouverte est une application $t \rightarrow u(t)$ d'un intervalle de temps dans l'espace des contrôles. Un contrôle en boucle fermée, appelé aussi une rétroaction, ou un bouclage, est une application $u \rightarrow g(t)$ définie sur les variables d'état du système. Un des objectifs de la théorie du contrôle est de déterminer des rétroactions qui stabilisent le système en un état particulier.

Bouclage statique

Définition 1.5. (Bouclage statique) On dit que u est un bouclage statique du système $\dot{x}(t) = f(x(t), u(t))$, $et(x(t) \in G, u(\cdot) \in U)$ si sa valeur $u(t)$ à l'instant t ne dépend que de $x(t)$, c'est-à-dire $u = g(x)$ où g est une fonction. Ce système s'écrit tout simplement

$$\dot{x}(t) = f(x, g(t)). \quad (1.21)$$

Il est représenté par la figure 1.1.

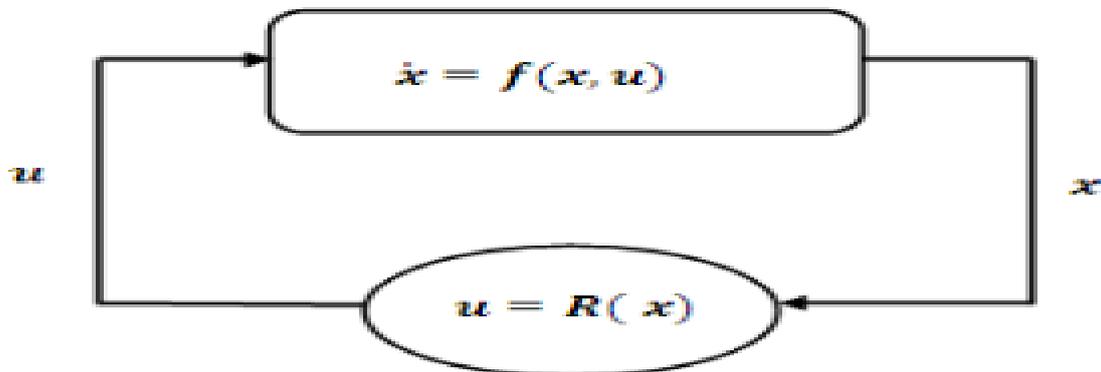


FIGURE 1.1 – bouclage

Le problème de la stabilité (ou régulation) consiste à maintenir le système près d'un équilibre x^* . Il s'agit donc de construire une loi de commande telle que x^* soit un équilibre asymptotiquement stable du système en boucle fermée (1.21).

Concepts de stabilité

On se donne un système

$$\dot{x}(t) = f(x). \quad (1.22)$$

tel que $f(0) = 0$, admettant $x = 0$ comme équilibre (noter que par un changement de variable on peut toujours ramener l'équilibre à l'origine).

Définition 1.6. L'équilibre $x = 0$ du système (1.21) est dit stable si pour tout $\epsilon > 0$, il existe $\eta > 0$ tel que pour toute solution $x(t)$ de (1.21), on ait

$$x(0) < \eta \Rightarrow \forall t \geq 0 \quad x(t) < \epsilon$$

Si l'équilibre n'est pas stable on dit qu'il est instable.

Définition 1.7. L'équilibre $x = 0$ du système (1.22) est dit attractif s'il existe $r > 0$ tel que pour toute solution $x(t)$ de (1.22) on ait

$$x(0) < r \Rightarrow \lim_{t \rightarrow \infty} x(t) = 0.$$

l'équilibre $x = 0$ du système (1.22) est dit globalement attractif si pour toute solution $x(t)$ de (1.22) on a $\lim_{t \rightarrow \infty} x(t) = 0$. L'ensemble B défini par la propriété

$$x(0) \in B \Rightarrow \lim_{t \rightarrow \infty} x(t) = 0.$$

s'appelle le bassin d'attraction de l'origine. Ainsi $x = 0$ est attractif si B est un voisinage de 0. Il est globalement attractif si $B = R^n$.

Définition 1.8. L'équilibre $x = 0$ du système (1.22) est dit asymptotiquement stable s'il est stable et attractif. Il est dit globalement asymptotiquement stable (GAS) s'il est stable et globalement attractif.

Définition 1.9. L'équilibre $x = 0$ du système (1.22) est dit exponentiellement stable s'il existe $r > 0$, $M > 0$ et $\alpha > 0$ tels que pour toute solution $x(t)$ on ait

$$\|x(t)\| < r \Rightarrow \|x(t)\| \leq M\|x(0)\|e^{-\alpha t}, \quad \text{pour tout } t > 0.$$

L'équilibre $x = 0$ du système (1.22) est dit globalement exponentiellement stable s'il existe $M > 0$ et $\alpha > 0$ tels que pour toute solution $x(t)$ de (1.22) on a

$$\|x(t)\| \leq M\|x(0)\|e^{-\alpha t}, \quad \text{pour tout } t > 0$$

On montre que, en générale d'après [29] stable n'implique pas attractif, attractif n'implique pas stable, exponentiellement stable implique asymptotiquement stable, et asymptotiquement stable n'implique pas exponentiellement stable

1.12 Observabilité

Soit le système linéaire autonome suivant :

$$\begin{cases} \dot{x}(t) = Ax(t) + Bu(t) \\ y(t) = Cx(t) + Du(t), \end{cases} \quad (1.23)$$

où $x(t) \in R^n, u(t) \in R^m, y(t) \in R^p, A \in M^n(R), B \in M_{n,m}(R), C \in M_{p,n}(R)$ et $D \in M_{p,m}(R)$. Dans toute la suite, on peut supposer que $D = 0$: cela ne change rien aux résultats qui suivent.

1.12.1 Définition et critères d'observabilité

Notons $(x_u(t, x_0), y_u(t, x_0))$ la solution de (1.23) tel que $x_u(0, x_0) = x_0$

Définition 1.10. Le système (1.23) est observable en temps t_f si

$$\forall x_1, x_2 \in R^n \implies \exists u \in L^\infty([0, t_f], R^m) / y_u(\cdot, x_1) \neq y_u(\cdot, x_2)$$

(Dans ce cas on dit que x_1 et x_2 sont distinguables).

Autrement dit, si x_1 et x_2 sont distinguables s'il existe un contrôle tel que les trajectoires observées diffèrent. De manière équivalente, on peut dire :

$$\forall x_1, x_2 \in R^n \forall u \in L^\infty([0, t_f], R^m) \quad y_u(\cdot, x_1) \neq y_u(\cdot, x_2) \implies x_1 = x_2,$$

i.e. la connaissance de la trajectoire observée détermine d'une manière univoque l'état initial. L'intérêt de la notion d'observabilité est le suivant. Si on considère le système comme une boîte noire à laquelle on applique une entrée (contrôle, input) $u(t)$, et de laquelle émerge une sortie (observable, output) $y(t)$, la propriété d'être distinguable signifie la possibilité de différencier par des expériences de type entrée-sortie. On est aussi motivé par la stabilisation. En effet, on a vu comment stabiliser un système par retour d'état. Or il peut s'avérer coûteux de mesurer l'état complet d'un système. On peut alors se demander si la connaissance partielle de cet état permet de reconstituer l'état complet (c'est la propriété d'observabilité), et de stabiliser le système entier : c'est la stabilisation par retour d'état dynamique, ou synthèse régulateur-observateur.

Théorème 1.7. [24] *le système (1.23) est observable (en temps t_f quelconque) si et seulement si :*

$$\text{rang} \begin{pmatrix} C \\ CA \\ \cdot \\ \cdot \\ \cdot \\ CA^{n-1} \end{pmatrix} = n$$

Preuve. Faisons une démonstration directe de ce théorème. on montre d'abord si le lemme fondamental suivant : □

Lemme 1.8. *Le système (1.23) est observable en temps t_f si et seulement si, pour le système observé $\dot{x} = Ax, y = Cx, x(0) = x_0$, et on a*

$$x_0 \neq 0 \implies y(\cdot) \equiv 0 \text{ sur } [0, t_f]$$

Preuv de Lemme :

Le système (1.23) est observable en temps T si et seulement si

$$(x_1 \neq x_2 \implies \exists u \in L^\infty([0, t_f], R^m) / y_u(\cdot, x_1) \neq y_u(\cdot, x_2) \text{ sur } [0, t_f])$$

$$(x_1 \neq x_2 \implies \exists u \in L^\infty([0, t_f], R^m) \exists t \in [0, t_f] /$$

\iff

$$Ce^{tA}x_1 + Ce^{tA} \int_0^t e^{-sA} Bu(s) ds \neq Ce^{tA}x_2 + Ce^{tA} \int_0^t e^{-sA} Bu(s) ds$$

$$\iff (x_0 = x_1 - x_2 \neq 0 \implies \exists t \in [0, t_f] / Ce^{tA}x_0 \neq 0)$$

$\iff (x_0 \neq 0 \implies y(\cdot) \equiv 0 \text{ sur } [0, t_f] \text{ pour le système } x = Ax, y = Cx, x(0) = x_0)$ On est maintenant en mesure de montrer le théorème. Si (1.23) n'est pas observable en temps t_f , alors

$$\exists x_0 \neq 0 / \forall t \in [0, t_f] y(t) = 0,$$

i.e

$$\forall t \in [0, t_f] Ce^{tA}x_0 = 0$$

D'où, par dérivations successives, et en prenant $t = 0$:

$$Cx_0 = CAx_0 = \dots = CA^{n-1}x_0 = 0$$

i.e

$$\begin{pmatrix} C \\ Ca \\ \vdots \\ CA^{n-1} \end{pmatrix} x_0 = 0 \quad \text{et Donc rang} \begin{pmatrix} C \\ CA \\ \vdots \\ CA^{n-1} \end{pmatrix} < n$$

Réciproquement, si le rang de cette matrice est strictement inférieure à n , alors il existe $x_0 \neq 0$ tel que

$$Cx_0 = CAx_0 = \dots = CA^{n-1}x_0 = 0$$

et donc par le théorème d'Hamilton-Cayley :

$$\forall t \in R Ce^{tA}x_0 = 0,$$

et par conséquent le système (1.23) n'est pas observable.

Remarque 1.5. Pour un système linéaire autonome, l'observabilité a lieu en temps quelconque si elle a lieu en temps t_f .

Remarque 1.6. La notion d'observabilité pour un système linéaire autonome ne dépend pas de la matrice B

Remarque 1.7. On a

$$\text{rang} \begin{pmatrix} C \\ CA \\ \vdots \\ CA^{n-1} \end{pmatrix} = n \iff \text{rang}(C^t \quad A^t C^t \quad \dots \quad [A^{n-1}]^t C^t) = n,$$

et par conséquent, le système $\dot{x} = Ax + Bu, y = Cx$ est observable si et seulement si le système $\dot{x} = A^t + C^t u$ est contrôlable. C'est la dualité contrôlabilité/observabilité. Ce fait, très important, permet de transférer aux systèmes observés tous les résultats établis sur les systèmes contrôlés.

aurait peut prouver cette équivalence directement en utilisant l'application entrée-sortie, et en remarquant qu'une application linéaire $E : L^2 \rightarrow R^n$ est surjective si et seulement si l'application adjointe $E^* : R^n \rightarrow L^2$ est injective.

Corollary 1.8.1. Le système (1.23) est observable en temps t_f si et seulement si la matrice

$$O(T) = \int_0^{t_f} e^{-sA} C^t C e^{-sA} ds$$

est inversible

Remarque 1.8. Dans le cas d'un système mono-sortie, la matrice d'observabilité est une matrice carrée

$$P = 1 \implies O(A, C) \in R^{n \times n}$$

Exemple 1.2. Considérons le système suivant :

$$\begin{cases} \dot{X}(t) = AX(t) + BU(t) \\ Y(t) = CX(t) + DU(t), \end{cases}$$

$$A = \begin{pmatrix} 1 & 2 & 1 \\ 3 & 2 & 1 \\ 1 & 1 & 1 \end{pmatrix} \quad B = \begin{pmatrix} 0 & 2 \\ 1 & 1 \\ 2 & 3 \end{pmatrix} \quad C = \begin{pmatrix} 1 & 0 & 1 \\ 1 & 2 & 1 \end{pmatrix} \quad D = 0$$

$n = 3$ états, $p = 2$ sorties

$$O(A, C) = \begin{pmatrix} C \\ CA \\ CA^2 \end{pmatrix} \quad O(A, C) \in R^{6 \times 3}$$

$$O(A, C) = \begin{pmatrix} 1 & 0 & 1 \\ 1 & 2 & 1 \\ 2 & 3 & 2 \\ 8 & 5 & 4 \\ 2 & 5 & 2 \\ 19 & 3 & 4 \end{pmatrix}$$

Matrice de rang 3 \implies système observable.

Exemple 1.3. Considérons le système suivant :

$$\begin{cases} \dot{X}(t) = AX(t) + BU(t) \\ Y(t) = CX(t) + DU(t), \end{cases}$$

Où

$$A = \begin{pmatrix} 1 & 1 \\ \alpha & \beta \end{pmatrix} B = \begin{pmatrix} 2 \\ 3 \end{pmatrix} C = \begin{pmatrix} 1 & 1 \end{pmatrix} D = 0$$

n = 2 états, p = 1 sortie

$$O(A, C) = \begin{pmatrix} C \\ CA \end{pmatrix} \quad O(A, C) \in R^{2 \times 2}$$

$$O(A, C) = \begin{pmatrix} 1 & 1 \\ \alpha & \beta \end{pmatrix} \quad \det O(A, C) = \beta - \alpha$$

Le système est observable si et seulement si le $\det O(A, C)$ est non nul, i.e.. $\beta \neq \alpha$.

Chapitre 2

Résolution d'un problème de contrôle optimal

2.1 Introduction

Les méthodes directes et les méthodes indirectes sont deux types de méthodes numériques par la résolution d'un problème de contrôle optimal.

Les méthodes directes consistent à discrétiser l'état et le contrôle, et réduisent le problème à un problème d'optimisation non linéaire. Les méthodes indirectes consistent à résoudre numériquement, par une méthode de tir, un problème aux valeurs limites obtenu par application du principe du maximum. Dans ce chapitre, on s'intéresse d'abord aux méthodes indirectes, puis aux méthodes directes.

2.2 Méthodes indirectes

Les méthodes indirectes sont basées sur le Principe du Maximum de Pontriaguine (PMP) qui donne une condition nécessaire d'optimalité. On cherche ensuite les trajectoires vérifiant ces conditions, et qui numériquement se ramènent à une méthode de tir. Le choix de ces méthodes s'explique par leurs avantages, la bonne rapidité de convergence et leur grande précision dans le traitement des problèmes de contrôle optimal mais lourdement dépendantes du point initial.

2.2.1 Principe du Maximum de Pontryagin

Avant d'énoncer le principe du maximum, introduisons certaines définitions et propriétés essentielles.

Définition 2.1. Un contrôle $u^0(t)$, $t \in [0, t_f]$ est dit optimal si $u^0(\cdot)$ est extrémal et $J(u^0(t)) < J(u(t))$ pour tout contrôle extrémal $u(t)$, $t \in [0, t_f]$.

Théorème 2.1. [30] *Considérons le système*

$$\forall t \in I, \dot{x}(t) = A(t)x(t) + B(t)u(t), x(0) = x_0$$

Supposons que le domaine de contraintes noté Ω est compact. Soit $t_f > 0$. Le contrôle u est extrémal sur $I = [0, t_f]$ si et seulement si il existe une solution non triviale $p(t)$, $t \in I$, de l'équation $\dot{p}(t) = -p(t)A(t)$ telle que :

$$p(t)B(t)u(t) = \max_{v \in \Omega} p(t)B(t)v, \quad (2.1)$$

pour presque tout $t \in [0, t_f]$

Définition 2.2. Le vecteur $p(t) \in R^n$ est appelé vecteur adjoint.

Définition 2.3. Le temps t_c auquel le contrôle extrémal $u(t)$, $t \in [0, t_f]$ change de signe est appelé temps de commutation.

Le théorème qui suit est l'énoncé général du principe du maximum de Pontryaguin.

Théorème 2.2. [1] *Considérons le système de contrôle dans R^n :*

$$\dot{x}(t) = f(t, x(t), u(t)), \quad (2.2)$$

Où $f : R \times R^n \times R^m \rightarrow R^n$ de classe C^1 , les contrôles sont des applications mesurables bornées à valeurs dans $\Omega \subset R^m$. Soit M_0 et M_1 deux ensembles de R^n . Notons par 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 .

On définit le coût

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

où $f : R \times R^n \times R^m \rightarrow R$ et $g : R \times R^n \rightarrow R$ sont de classe C_1 , $x(\cdot)$ est la solution de (2.2) associée au contrôle u .

On considère le problème de contrôle optimal suivant : Déterminer une trajectoire reliant M_0 à M_1 en minimisant le coût J . Le temps final peut être fixé ou non. 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, appelé vecteur adjoint, et un réel $p_0 \leq 0$ tel que le couple $(p(\cdot), p_0)$ est non trivial et tels que pour presque tout $t \in [0, T]$,

$$\dot{x}(t) = \frac{\partial H}{\partial p}(t, x(t), p(t), p^0, u(t)), \quad (2.3)$$

$$\dot{p}(t) = -\frac{\partial H}{\partial x}(t, x(t), p(t), p^0, u(t)), \quad (2.4)$$

où $H(t, x, p, p^0, u) = p'(t)f(t, x, u) + p^0 f^0(t, x, u)$ est le Hamiltonien du système, et on a la condition de maximisation presque partout sur $[0, t_f]$

$$H(t, x, p, p^0, u(t)) = \max_{v \in U} H(t, x(t), p(t), p^0, v). \quad (2.5)$$

Si de plus M_0 et M_1 (ou juste l'un des deux ensembles) sont des variétés de R^n ayant des espaces tangent en $x(0) = x_0 \in M_0$ et $x(T) = x_1 \in M_1$, alors le vecteur adjoint peut être construit de manière à vérifier les conditions de transversalités aux deux extrémités (ou juste l'une des deux)

$$p(0) \perp T_{x(0)}M_0 \quad (2.6)$$

$$\left(p(T) - \dot{p} \frac{dg}{dx}(T, x(T)) \right) \perp T_{x(T)}M_1 \quad (2.7)$$

Remarque 2.1. Si f et f^0 ne dépendent pas du temps t c'est à dire si le système considéré est autonome, alors l'Hamiltonien H ne dépend pas de t et on a :

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

Remarque 2.2. La convention $p_0 \leq 0$ conduit au principe du maximum. La condition $p_0 \geq 0$ conduira au principe du minimum.

Remarque 2.3. L'orsqu'il n'y a pas de contraintes sur le contrôle, la condition de maximum (2.5) devient $\frac{\partial H}{\partial u} = 0$.

Définition 2.4. Les conditions (2.6) et (2.7) sont appelées conditions de transversalité sur le vecteur adjoint

2.2.2 Méthode de tir simple :

La méthode de tir simple est basée sur le Principe du Maximum de Pontryagin (PMP). Elle consiste à trouver un zéro de la fonction de tir associée au problème original. Il s'agit d'une méthode rapide et de haute précision, qui ne requiert pas d'hypothèses sur la structure du contrôle.

Considérons le problème du contrôle optimal suivant :

$$\begin{cases} \min J(u(t_f), t_f) = \int_0^{t_f} f^0(x(t), u(t), t)dt + g(t_f, x(t_f)) \\ \dot{x}(t) = f(x(t), u(t), t), \quad t \in T = [0, t_f] \\ x(0) = x_0, \quad p(0) = p_0 \\ x(t_f) = x_1, \quad p(t_f) = p_1 \end{cases} \quad (2.1)$$

et supposons dans un premier temps que le temps final est fixé. Le principe du maximum donne une condition nécessaire d'optimalité et affirme que toute trajectoire optimale est la projection d'une extrémale. Si l'on est capable, à partir de la condition de maximum, d'exprimer le contrôle extrémal en fonction de $(x(t), p(t))$ alors le système extrémal est un système différentiel de la forme $\dot{z}(t) = F(t, z(t))$ où $z(t) = (x(t), p(t))$ et les conditions initiales, finales et les conditions de transversalité, se mettent sous la forme :

$$R(z(0), z(t_f)) = R(x_0, p_0, x_1, p_1) = 0$$

Finalement, on obtient le problème aux valeurs limites suivant :

$$\begin{cases} \dot{z}(t) = F(t, z(t)) \\ R(z(0), z(t_f)) = 0 \end{cases} \quad (2.2)$$

Notons $z(t, z_0)$ la solution du problème de Cauchy suivant :

$$\dot{z} = F(t, z(t)), z(0) = z_0.$$

La fonction de tir est définie par :

$$G(t_f, z_0) = R(z_0, z(t_f, z_0)).$$

Le problème (2.2) aux valeurs limites est alors équivalent à la résolution du système suivant :

$$G(t_f, z_0) = 0,$$

Il s'agit de déterminer un zéro de la fonction G . Ceci peut se résoudre par la méthode de Newton.

2.2.3 Méthode de tir multiple :

Par rapport à la méthode de tir simple, la méthode de tir multiple découpe l'intervalle $[0, t_f]$ en N intervalles $[t_i, t_{i+1}]$, et se donne comme inconnues les valeurs $z(t_i)$ au début de chaque sous-intervalle. Il faut prendre en compte des conditions de recollement en chaque temps t_i (conditions de continuité). L'intérêt est d'améliorer la stabilité de la méthode. Une référence classique pour l'algorithme de tir multiple est [9]. L'application du principe du maximum réduit le problème à un problème aux valeurs limites du type

$$\dot{z}(t) = F(t, z(t)) = \begin{cases} F_0(t, z(t)) & \text{si } t_0 \leq t < t_1 \\ F_1(t, z(t)) & \text{si } t_1 \leq t < t_2 \\ \vdots \\ F_s(t, z(t)) & \text{si } t_s \leq t \leq t_f \end{cases} \quad (2.10)$$

où $z = (x, p) \in R^{2n}$ (p est le vecteur adjoint), et $t_1, t_2, \dots, t_s \in [t_0, t_f]$ peuvent être des temps de commutation ; dans le cas où le problème inclut des contraintes sur l'état, ce peut être des temps de jonction avec un arc frontière, ou bien des temps de contact avec la frontière. On a de plus des conditions de continuité sur l'état et le vecteur adjoint aux points de commutation. Dans le cas de contraintes sur l'état, on a des conditions de saut sur le vecteur adjoint, et des conditions sur la contrainte C en des points de jonction ou de contact (voir à ce sujet [17]). De plus on a des conditions aux limites sur l'état, le vecteur adjoint (conditions de transversalité), et sur le Hamiltonien si le temps final est libre

Remarque 2.4. A priori le temps final t_f est inconnu. Par ailleurs dans la méthode de tir multiple le nombre s de commutations doit être fixé ; on le détermine lorsque c'est possible par une analyse géométrique du problème.

La méthode de tir multiple consiste à subdiviser l'intervalle $[t_0, t_f]$ en N sousintervalles, la valeur de $z(t)$ au début de chaque sous-intervalle étant inconnue. Plus précisément, soit $t_0 < \gamma_1 < \dots < \gamma_k < t_f$ une subdivision fixée de l'intervalle $[0, t_f]$. En tout point γ_j la fonction z est continue. On peut considérer γ_j comme un point de commutation fixe, en lequel on a

$$\begin{aligned} z(\gamma_j^+) &= z(\gamma_j^-) \\ \gamma_j &= \gamma_j^* \end{aligned}$$

On définit maintenant les noeuds

$$\{\tau_1, \dots, \tau_m\} = \{0, t_f\} \cup \{\gamma_1, \dots, \gamma_k\} \cup \{t_1, \dots, t_s\}.$$

Finalement on est conduit au problème aux valeurs limites

$$\bullet \dot{z}(t) = F(t, z(t)) = \begin{cases} F_1(t, z(t)) & \text{si } \tau_1 \leq t < \tau_2 \\ F_2(t, z(t)) & \text{si } \tau_2 \leq t < \tau_3 \\ \vdots & \\ F_{m-1}(t, z(t)) & \text{si } \tau_{m-1} \leq t \leq \tau_m \end{cases}$$

$$\forall j \in \{2, \dots, m-1\} \quad r_j(T_j, z(T_j^-), z(T_j^+)) = 0$$

$$r_m(T_m, z(T_1), z(T_m)) = 0$$

où $\tau_1 = t_0$ est fixé, $\tau_m = t_f$, et les r_j représentent les conditions intérieures ou limites précédentes

Remarque 2.5. On améliore la stabilité de la méthode en augmentant le nombre de noeuds. C'est là en effet le principe de la méthode de tir multiple, par opposition

à la méthode de tir simple où les erreurs par rapport à la condition initiale évoluent exponentiellement en fonction de $t_f - t_0$ [9]. Bien sûr dans la méthode de tir multiple il y a beaucoup plus d'inconnues que dans la méthode de tir simple, mais éventuellement l'intégration du système (2.10) peut se paralléliser. Posons $z_j^+ = z(T_j^+)$, et soit $z(t, T_{j-1}, z_{j-1})$ la solution du problème de Cauchy

$$z'(t) = F(t, z(t)), z(T_{j-1}) = z_{j-1}^+$$

On a

$$z(T_j^-) = z(T_j^-, T_{j-1}, z(T_{j-1}^+))$$

Les conditions intérieures et frontières s'écrivent

$$\forall j \in (2, \dots, m-1) \quad r_j(T_j, z(T_j^-, T_{j-1}, z(T_{j-1}^+)), z_j^+) = 0 \quad (2.11)$$

$$r_m(T_m, z_1^+, z(T_m^-, z_{m-1}^+), z_{m-1}^+) = 0$$

Posons maintenant

$$Z = (z_1^+, T_m, z_2^+, T_2, z_{m-1}^+, T_{m-1})^T \in R^{(2n+1)(m-1)}$$

(ou $z \in R^{2n}$) Alors les conditions (2.11) sont vérifiées si

On s'est donc ramené à déterminer un zéro de la fonction G , qui est définie sur un espace vectoriel dont la dimension est proportionnelle au nombre de points de commutation et de points de la subdivision. L'équation $G = 0$ peut alors être résolue itérativement par une méthode de type Newton

2.2.4 Rappels sur les méthodes de Newton

Il s'agit de résoudre numériquement $G(z) = 0$, Ou $G : R^p \rightarrow R^p$ est une fonction de classe C^1 L'idée de base est la suivante. Si Z^k est proche d'un zéro z de G , alors

$$0 = G(z) = G(z_k) + dG(z_k).(z - z_k) + o(z - z_k)$$

On est alors amené à considérer la suite définie par récurrence

$$z_{k+1} = z_k - (dG(z_k))^{-1}.G(z_k)$$

un point initial $z_0 \in R^p$ étant choisi, et on espère que z_k converge vers le zéro z . Ceci suppose donc le calcul de l'inverse de la matrice jacobienne de G , ce qui doit être évité numériquement. Il s'agit alors, à chaque étape, de résoudre l'équation

$$G(z_k) + dG(z_k).d_k = 0$$

où d_k est appelé direction de descente, et on pose z_{k+1}

Sous des hypothèses générales, l'algorithme de Newton converge, et la convergence est quadratique. Cette méthode permet, en général, une détermination très précise d'un zéro. Son inconvénient principal est la petitesse du domaine de convergence. Pour faire converger la méthode, il faut que le point initial z_0 soit suffisamment proche de la solution recherchée z . Ceci suppose donc que pour déterminer le zéro z il faut avoir au préalable une idée approximative de la valeur de z . Du point de vue du contrôle optimal, cela signifie que, pour appliquer une méthode de tir, il faut avoir une idée a priori de la trajectoire optimale recherchée. Ceci peut sembler paradoxal, mais il existe des moyens de se donner une approximation, même grossière, de cette trajectoire optimale. Il s'agit là en tout cas d'une caractéristique majeure des méthodes de tir : elles sont très précises mais requièrent une connaissance a priori (plus ou moins grossière) de la trajectoire optimale recherchée.

2.3 Méthodes directes

Les méthodes directes consistent à transformer le problème de contrôle optimal en un problème d'optimisation non linéaire en dimension finie

2.3.1 Présentation - formulation mathématique

2.3.2 Discrétisation totale "Tir direct" :

C'est la méthode la plus évidente lorsqu'on aborde un problème de contrôle optimal. En discrétisant l'état et le contrôle, on se ramène à un problème d'optimisation non linéaire en dimension finie de la forme :

$$\min_{z \in C} F(Z)$$

où

$$Z = (x_1, \dots, x_N, u_1, \dots, u_n), \text{ et } C = \{Z \mid g_i(Z) = 0, i \in 1, \dots, r, g_j(Z) \leq 0, j \in r + 1, \dots, m\}$$

Plus précisément, la méthode consiste à choisir les contrôles dans un espace de dimension finie, et à utiliser une méthode d'intégration numérique des équations différentielles. Considérons donc une subdivision $0 = t_0 < t_1 < \dots < t_N = t_f$ de l'intervalle $[0, t_f]$. Par ailleurs, choisissons une discrétisation de l'équation différentielle, par exemple choisissons ici (pour simplifier) la méthode d'Euler (voir annexe A). On obtient alors, en posant $h_i = t_{i+1} - t_i$

$$x_{i+1} = x_i + h_i f(t_i, x_i, u_i).$$

Remarque 2.6. Il existe une infinité de variantes. D'une part, on peut discrétiser l'ensemble des contrôles admissibles par des contrôles constants par morceaux, ou affines par morceaux, ou des splines, etc. D'autre part, il existe de nombreuses méthodes pour discrétiser une équation différentielle ordinaire : méthode d'Euler , Runge-Kutta (Voir annexe B). Le choix de la méthode dépend du problème abordé. La discrétisation précédente conduit donc au problème de programmation non linéaire

$$x_{i+1} = x_i + h_i f(t_i, x_i, u_i), i = 0, \dots, N - 1,$$

$$\min C(x_0, \dots, x_N, u_0, \dots, u_N)$$

,

$$u_i \in \Omega, i = 0, \dots, N - 1,$$

i.e. un problème du type (*).

Remarque 2.7. Cette méthode est très simple à mettre en oeuvre. De plus l'introduction d'éventuelles contraintes sur l'état ne pose aucun problème. D'un point de vue plus général, cela revient à choisir une discrétisation des contrôles, ainsi que de l'état, dans certains espaces de dimension finie :

$$u \in Vect(U_1, \dots, U_N), \text{ i.e. } u(t) = \sum_1^N u_i U_i(t), \quad u_i \in R,$$

$$x \in Vect(X_1, \dots, X_N), \text{ i.e. } x(t) = \sum_1^N x_i X_i(t), \quad x_i \in R,$$

où les $U_i(t)$ et $X_i(t)$ représentent une base de Galerkin (Voir annexe H). Typiquement, on peut choisir des approximations polynomiales par morceaux. L'équation différentielle, ainsi que les éventuelles contraintes sur l'état ou le contrôle, ne sont vérifiées que sur les points de la discrétisation. On se ramène bien à un problème d'optimisation non linéaire en dimension finie de la forme (*). La résolution numérique d'un problème de programmation non linéaire du type (*) est standard. Elle peut être effectuée, par exemple, par une méthode de pénalisation (Voir annexe F), ou par une méthode SQP (sequential quadratic programming) (Voir annexe D). Dans ces méthodes, le but est de se ramener à des sous-problèmes plus simples, sans contraintes, en utilisant des fonctions de pénalisation pour les contraintes, ou bien d'appliquer les conditions nécessaires de Kuhn-Tucker pour des problèmes d'optimisation avec contraintes. Pour le problème (*), (**), les conditions de Kuhn-Tucker s'écrivent

$$\nabla F(z) + \sum_i^m \lambda_i \nabla g_i(Z) = 0,$$

où les multiplicateurs de Lagrange (Voir annexe E) λ_i vérifient

$$\lambda_i g_i(Z) = 0, i \in 1, \dots, r, \text{ et } \lambda_i > 0, i \in r + 1, \dots, m.$$

Les méthodes SQP consistent à calculer de manière itérative ces multiplicateurs de Lagrange, en utilisant des méthodes de Newton. A chaque itération, on utilise une méthode de quasi-Newton pour estimer le Hessien du Lagrangien associé au problème de programmation non linéaire, et on résout un sous-problème de programmation quadratique basé sur une approximation quadratique du Lagrangien. Pour plus de détails sur cette méthode[18]

2.4 les caractéristiques des méthodes directes et indirectes.

2.4.1 Méthode directe :

- peu sensibles au choix de la condition initiale
- contrôles (globalement) optimaux en boucle fermée
- mise en oeuvre simple, sans connaissance a priori
- précision numérique basse ou moyenne
- problème des minima locaux
- efficaces en basse dimension
- gourmandise en mémoire

2.4.2 Méthode indirecte :

- connaissance a priori de la structure de la trajectoire optimale
- très sensibles au choix de la condition initiale
- contrôles (localement) optimaux en boucle ouverte
- efficaces en toute dimension
- petit domaine de convergence
- difficulté théorique de la prise en compte de contraintes sur l'état
- très grande précision numérique
- calculs parallélisables

Exemple 2.1. soit le problème :

$$\begin{aligned} & \min \int_0^2 u^2 dt \\ \dot{x} &= y \quad x(0) = 0, \quad y(0) = 0 \\ \dot{y} &= u \quad x(2) = 0.5, \quad y(2) = 0 \end{aligned}$$

avec : $tf = 2$

La fonction Hamiltonien :

$$H = p^0 f^0 + p'(t)f(t, x, u) = -u^2 + (p_1 \ p_2) \begin{pmatrix} y \\ u \end{pmatrix} = -u^2 + p_1 y + p_2 u$$

Le principe du maximum :

$$\max_{|u| \leq 1} H = \max(p^0 f^0 + p'(t)f(t, x, u)) = -u^2 + (p_1 \ p_2) \begin{pmatrix} y \\ u \end{pmatrix} = -u^2 + P_1 y + p_2 y$$

$$\frac{\partial H}{\partial u} = -2u(t) + p_2 = 0 \rightarrow u(t) = \frac{1}{2}p_2$$

donc :

si $|\frac{1}{2}p_2| < 1$ alors $u^* = \frac{1}{2}p_2$

si $|\frac{1}{2}p_2| \geq 1$ alors $u^* = \text{signe}(p_2)$

Les vecteurs adjoints :

$$\dot{p}_1 = -\frac{\partial H}{\partial x} = 0 \Rightarrow p_1(t) = C_0$$

$$\dot{p}_2 = -\frac{\partial H}{\partial y} = -p_1(t) \Rightarrow p_2(t) = -c_0 t + c_1$$

$$\text{si } |\frac{1}{2}p_2| < 1 \text{ alors } u^* = \frac{1}{2}p_2$$

$$\text{si } |\frac{1}{2}p_2| \geq 1 \text{ alors } u^* = \text{signe}(p_2)$$

Application de la méthode tir simple :

Nous avons le problème aux deux bouts :

$$(PDB) = \begin{cases} \dot{x}(t) = y(t) \\ \dot{y}(t) = u(t) \\ \dot{p}_x(t) = 0 \\ \dot{p}_y(t) = -p_x(t) \\ x(0) = 0; y(0) = 0 \\ x(tf) = 0.5; y(tf) = 0 \end{cases}$$

En posons :

$$z(t) = (x(t); y(t); p_x(t); p_y(t)) = (z_1(t); z_2(t); z_3(t); z_4(t))$$

→

$$\begin{cases} \dot{z}_1(t) = z_2(t) \\ \dot{z}_2(t) = u(t) \\ \dot{z}_3(t) = 0 \\ \dot{z}_4(t) = -z_3(t) \\ z_1(0) = 0; z_2(0) = 0 \\ z_3(0) = c_1 \in \mathbf{R}; z_4(0) = c_2 \in \mathbf{R} \end{cases}$$

on pose $\mathbf{Z} = \begin{pmatrix} z_1 \\ z_2 \\ z_3 \\ z_4 \end{pmatrix}$

Donc le problème aux deux bouts est équivalent :

$$\left\{ \begin{array}{l} \dot{Z} = F(t, Z), \quad Z(0) = \begin{pmatrix} 0 \\ 0 \\ c_1 \\ c_2 \end{pmatrix} \quad Z(tf) = \begin{pmatrix} 0.5 \\ 0 \\ c_3 \\ c_4 \end{pmatrix} \end{array} \right.$$

Le problème de Cauchy associée au problème est :

$$\left\{ \begin{array}{l} \dot{Z} = F(t, Z) \\ Z(0) = h \end{array} \right.$$

La fonction de tir est défini par :

$$G(h) = Z(tf, h) - Z(tf)$$

La connaissance de la fonction de tir permet de considérer que ce problème optimal est entièrement résolu mathématiquement .

Résoudre le problème (PDB) est équivalent à chercher un zéro de l'équation $G(h) = 0$ c'est à dire chercher : $h/G(h) = 0$

Résolution avec Matlab

function direct

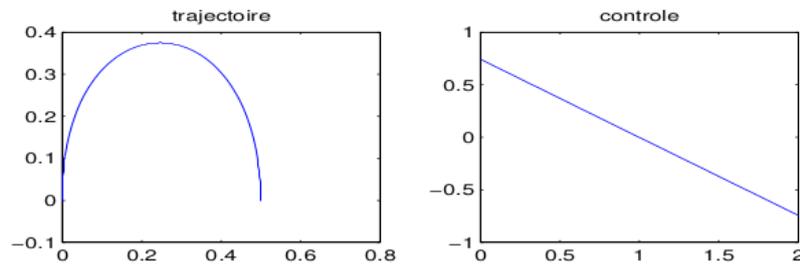
```
%discritisation directe (en utilisant fmincon.m du problème de temps minimale)
%xdot=y, ydot=u, |u| <= 1,
%le problème étant de joindre (0,0) a (0.5,0)en temps minimal.
clear all ; close all ; clc;
global tf
N=100 ; %nombre de pas de discritisation
uinit=2*rand(N,1)-1; %initialisation aléatoire du contrôle
%point de départ pour fmincon
lb=-ones(N,1);
ub=ones(N,1);
%contrainte sur le contrôle |u| <= 1
%programme principale
[rep,Fval,exitflag]=fmincon(@energie,uinit,[],[],[],[],lb,ub,@cond);
```

```

exitflag
x(1)=0 ;
y(1)=0 ;
for i=1:N
x(i+1)=x(i)+2/N*y(i);
y(i+1)=y(i)+2/N*rep(i);
end
%calcul de la trajectoire optimale
subplot(121); plot(x,y);axis square;
title('trajectoire');
subplot(122); plot(linspace(0,tf,N),rep(1:N));
axis square; title('contrôle');
%-----
function [c,ceq]= cond(u) global tf ;
N=length(u);
c=0;
xf=0;
yf=0;
for i=1:N
xf=xf+tf/N*yf;%calcul de point final au temps tf
yf=yf+tf/N*u(i); %avec la méthode d'Euler explicite
end
ceq=[xf-0.5;yf ];
%on impose la condition finale xf=0,yf=-1
%-----
function val = energie(u)
global tf;
N=length(u);
val=0;
for i=1:N
val=val+tf/N* u(i)*u(i);
end

```

Les résultat sont tracer sur la figure suivante :



Résultat de la méthode directe

Sur cette figure on a 2 graphes :

Le premier représente la trajectoire qui est varié de 0 à 0.39.

Le deuxième graphe représente le contrôle qui est varié de -0.6 à 0.6 .

Les deux fonctions sont cassées.

function tirsimple

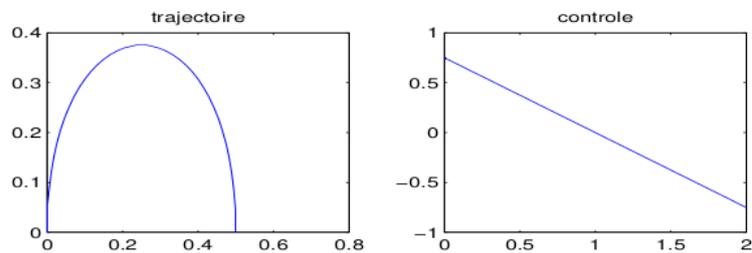
```
%Méthode de tir simple, en utilisant fsolve, pour le système de contrôle
%xdot=y, ydot=u, |u| <= 1;
%on veut aller de (0,0) a (0.5,0)en temps minimal
clear all; clf; clc; format long;
global x0 tf
X0=[0;0];
p0=[0.5;0.5];
%calcul de p0 ;
options = optimset('Display','iter','largescale','on');
%programme principale
[p0tf,FVAL,EXITFLAG]=fsolve(@F,p0,options);
EXITFLAG
%1 si la méthode converge, -1 sinon
%Trace de la trajectoire optimale
options= odeset('AbsTol',1e-9,'RelTol',1e-9);
[t,z]=ode45(@sys,[0;tf],[x0;p0tf],options);
subplot(121);plot(z(:,1),z(:,2));
axis square;title('trajectoire');
if abs(z(:,4)) < 2
z(:,5)= -z(:,4)/2 ;
else
z(:,5)=-sign(z(:,4));
end
```

```

subplot(122);plot(t,z(:,5));
axis square; title('contrôle');
%-----
function Xzero=F(X)
%définition de la fonction dont on cherche un zéro
global x0 tf ;
options = odeset('AbsTol',1e-9,'RelTol',1e-9);
[t,z]=ode113(@sys,[0;tf ],[x0;X],options);
Xzero =
[ z(end,1)-0.5
z(end,2)];
%on impose xf=0.5
%on impose yf=0
%-----
function zdot =sys(t,z)
if abs(z(4)) < 2
u= -z(4)/2 ;
else
u=-sign(z(4));
end
zdot=
[z(2)
u
0
-z(3)];% système extrêmeal

```

Les résultats sont tracés sur la figure suivante :



Résultat de la méthode de tir simple

Dans cette figure on a deux graphes le premier représente la trajectoire qui est variée entre 0 et 0.39.

Le deuxième représente le contrôle qui est varié de -0.6 à 0.6 ;
qui sont deux fonctions lisses.

Remarque 2.8. Si on compare les résultats de la méthode directe et celle de la méthode indirecte dans l'exemple on trouve que les méthodes indirectes sont plus précises

2.5 Conclusion

De la comparaison on peut conclure ce qui suit :
Il serait préférable d'opter pour une méthode indirecte, si on a besoin de calculer la trajectoire optimale de manière très précise.
On utilisera plutôt une méthode directe, si on n'a pas besoin d'une grande précision de calcul, donc les méthodes directes permettent l'obtention d'une solution approchée.

Chapitre 3

Temps minimum d'une masse avec limite de force

Introuction

Dans ce chapitre on présentera une implémentation des notions du chapitre précédent sur le problème de contrôle d'une masse avec limite de force. Les résultats obtenus seront ensuite comparés à ceux obtenus en programmant les deux méthodes sur MATLAB.

On considère le problème d'un véhicule se déplaçant en ligne droite sans frottement :

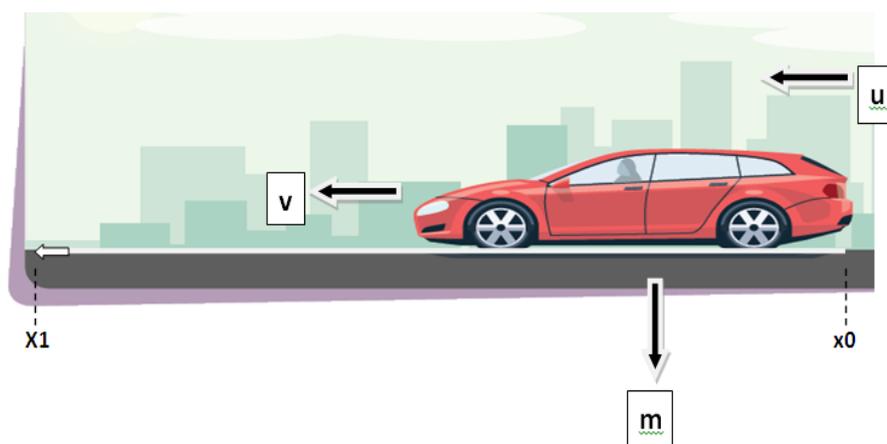


FIGURE 3.1 – véhicule déplaçant sur une ligne droite

3.1 Rappels sur les lois de la physique(Newton)

Les lois du mouvement de Newton sont les trois principes fondamentaux concernant le mouvement des corps, on cite dans ce chapitre :

3.1.1 La première loi de Newton

Dans un référentiel Galiléen ; si un corps est au repos ou s'il est en mouvement rectiligne uniforme, les forces qu'il subit se compensent (la somme des forces égale à zéro)

$$\sum F = 0$$

3.1.2 La deuxième loi de Newton

Cette loi est appelée aussi principe fondamental de la dynamique. Quand une force résultante s'exerce sur un objet, celui-ci est soumis à une accélération qui a la même direction que la force :

$$\sum \vec{F} = m\vec{\eta}$$

où m : masse d'inertie de l'objet.

Le mouvement est décrit par un système d'équations différentielles de 2^eme ordre.

$$m\ddot{y}(t) = f(t) \quad (3.1)$$

où m est la masse, $y(t)$, $\dot{y}(t)$ et $\ddot{y}(t)$ sont respectivement la position, la vitesse, l'accélération et $f(t)$ est la force externe appliquée au système. Posons

$$x_1(t) = y(t); x_2(t) = \dot{y}(t) \quad (3.2)$$

Par conséquent, (3.1) s'écrit sous forme d'un système d'équations différentielles d'ordre 1

$$\begin{cases} \dot{x}_1(t) = x_2(t) \\ \dot{x}_2(t) = u(t) \end{cases} \quad (3.3)$$

où $u(t) = f(t)/m$

On suppose que la commande $u(t)$ vérifie

$$|u(t)| \leq 1 ; t \in [t_0, t_f] \quad (3.4)$$

on obtient le système ci-dessous

$$(P) = \begin{cases} \min J(u(t)) = \int_{t_0}^{t_f} 1 dt \\ \dot{x}_1(t) = x_2(t) \\ \dot{x}_2(t) = u(t) \\ |u(t)| \leq 1; t \in [t_0, t_f] \\ x_1(0) = 0, x_1(t_f) = x_{1f} \\ x_2(0) = 0, x_2(t_f) = x_{2f} \end{cases} \quad (3.5)$$

3.2 Contrôlabilité

Comme le système (P) est linéaire, alors il s'écrit sous la forme matricielle suivante :

$$\dot{X} = Ax + Bu$$

Où :

$$A = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix} \quad \text{et} \quad B = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

nous avons $n = 2$ états, $m = 1$ entrée.

Et la matrice de commandabilité du système est :

$$K(A, B) = [B \ AB] \in \mathbb{R}^{2 \times 2}$$

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

La matrice K est de rang $2 = n$, d'après le théorème de Kalman, le système (3.5) est contrôlable.

3.3 Résolution théorique

Application du Principe du maximum de Pontryagin

Calcul de l'hamiltonien :

La fonction hamiltonienne de ce problème s'écrit :

$$H(x(t), u(t), p(t), p^0) = 1 + p_1(t)x_2 + p_2(t)u(t) + p^0 \quad (3.6)$$

où p^0 est une constante négative qui est la variable duale du coût et $p(t) \in \mathbb{R}^2$ est le vecteur adjoint, solution du système :

$$\begin{cases} \dot{p}_1(t) = -\frac{\partial H}{\partial x_1^*} \\ \dot{p}_2(t) = -\frac{\partial H}{\partial x_2^*} \end{cases} \quad (3.7)$$

On en déduit que :

$$\begin{cases} p_1^*(t) = K_1 \\ p_2^*(t) = K_2 - K_1 t \end{cases} \quad (3.8)$$

avec K_1 et K_2 sont des constante $\in \mathbb{R}$

Donc

$$H = K_1 x_2 + (K_2 - K_1 t)u$$

Principe du maximum :

Par ailleurs la condition du maximum (du principe du maximum) :

$$\max_{|u(t)| \leq 1} H = K_1 x_2 + \max_{|u(t)| \leq 1} (K_2 - K_1 t)u$$

nous donne :

$$u(t) = \text{signe}(K_2 - K_1 t). \quad (3,9)$$

Comme la droite $(k_2 - k_1 t)$ change au plus une fois de signe sur l'intervalle $t_0 \leq t \leq t_1$ alors, chaque commande optimale $u(t)$, $t_0 \leq t \leq t_1$ est une fonction constante par morceaux, prenant les valeurs 1 et admettant au plus deux intervalles de constance. En particulier les contrôles extrémaux ont au plus une commutation. Inversement, une telle fonction $u(t)$ peut être tirée de la relation (3,9) pour certaines valeurs de constantes k_1, k_2

i. $u = +1$: la résolution de système du problème (3.8), donne :

$$\dot{x}_2 = u = +1 \Rightarrow x_2 = t + s_2$$

et

$$\dot{x}_1 = x_2 = t + s_2 \Rightarrow x = \frac{1}{2}t^2 + s_2t + s_2 = \frac{1}{2}(t + s_2)^2 + (s_1 - \frac{s_2^2}{2})$$

où s_1, s_2 sont des constantes d'intégration. De là on obtient x_1 en fonction de x_2 :

$$x_1 = \frac{1}{2}y^2 + s \quad (3, 10)$$

où $s = s_1 - \frac{1}{2}s_2^2$ est une constante. Par conséquent la portion de trajectoire de phase pour laquelle $u = +1$ est un arc de parabole (3.10).

De la même manière :

ii. Pour $u = -1$, on obtient les trajectoires suivantes :

$$x_2 = -t + z_2$$

et

$$x_1 = \frac{-1}{2}t^2 + z_2t + z_2 = \frac{-1}{2}(-t + z_2)^2 + (z_1 + \frac{z_2^2}{2})$$

où z_1, z_2 sont des constantes d'intégration.

Alors on obtient x_1 en fonction de x_2 :

$$x_1 = \frac{-1}{2}x_2^2 + z \quad (3.11)$$

Les points représentatifs parcourent les paraboles (3.11) de bas en haut (car $\dot{x}_2 = +1$) et les paraboles (3.11) de haut en bas (car $\dot{x}_2 = -1$).

iii. Si la commande $u(t) = +1$ et ensuite $u(t) = -1$, la trajectoire de phase est composée de deux portions de paraboles contigües, la deuxième portion est située sur la parabole (3,11) qui passe par l'origine (car la trajectoire cherchée doit finir à l'origine).

iv. Si la commande $u(t) = -1$ et ensuite $u(t) = +1$, la trajectoire de phase est remplacée par une trajectoire qui lui est symétrique par rapport au centre.

La solution du problème optimal de cet exemple peut être interprétée de la manière suivante :

Soit $v(x_1, x_2) = v(z)$, la fonction d'efinie dans le plan (x_1, x_2) comme suit :

$$v(z) = \begin{cases} +1, & \text{au dessous de la ligne } AOB \text{ et sur l'arc } AO \\ -1, & \text{au dessus de la ligne } AOB \text{ et sur l'arc } BO. \end{cases} \quad (3.9)$$

Sur chaque trajectoire optimale, la valeur $u(t)$ (à l'instant t) est égale à $v(x(t))$. Donc, le système (3.5) devient en remplaçant u par la fonction $v(x)$:

$$\begin{cases} \dot{x}_1(t) = x_2(t) \\ \dot{x}_2(t) = v(x_1, x_2) \\ |u(t)| \leq 1 \\ x_1(0) = 0 & x_1(t_f) = 0 \\ x_2(0) = 0 & x_2(t_f) = -1 \end{cases}$$

La solution du système (3.9) (avec un état initial $x_0 = (0, 0)$) nous donne la trajectoire de phase optimale qui mène à un état final arbitraire x_1

En d'autre terme, le système (3.9) est un système d'équations différentielles (avec un deuxième ordre discontinu) dont la résolution donne les trajectoires optimales. Décrivons la trajectoire obtenue en prenant $u(t) = 1$ sur $[0, t_1[$, puis $u(t) = -1$ sur $]t_1, T]$. D'après les équations (3.9), on obtient

- si $0 \leq t \leq t_1$, Alors $x_1(t) = \frac{t^2}{2}$ et $x_2(t) = t$ - si $t_1 \leq t \leq T$ Alors $x_1(t) = -\frac{t^2}{2} + 2t_1t - t_1^2$ et $x_2(t) = -t + 2t_1$ Les trajectoires obtenues en prenant d'abord $u = -1$, puis $u = 1$, sont les symétriques des précédentes par rapport à l'origine

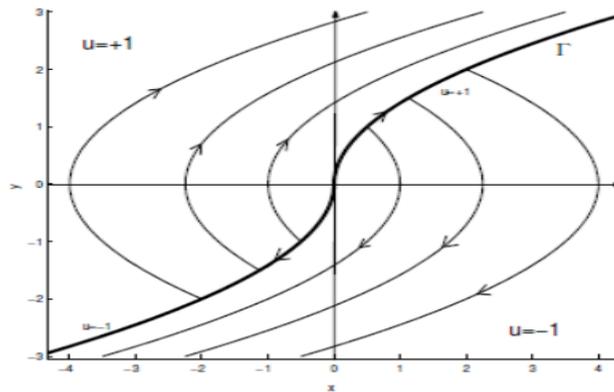


FIGURE 3.2 – Synthèse optimale

Il est clair que la courbe γ définie par

$$\gamma = \{(x_1, x_2) \in \mathbb{R}^2 \mid x_1 = \frac{x_2^2}{2} \text{ signe}(x_2)\}$$

est la courbe de commutation. Plus précisément, le contrôle temps minimal est donné par

- Si $x_1 > \frac{x_2^2}{2} \text{ signe}(x_2)$ ou si $x_1 = \frac{x_2^2}{2}$, alors $u(x_1, x_2) = +1$
- Si $x_1 < \frac{x_2^2}{2} \text{ signe}(x_2)$ ou si $x_1 = -\frac{x_2^2}{2}$, alors $u(x_1, x_2) = -1$

Calculons la fonction de temps minimal $t_f(x_1, x_2)$ pour aller de $(0, 0)$ à (x_1, x_2) . Supposons que le point (x_1, x_2) est en dessous de la courbe γ . Ce point est atteint par la succession d'un arc $u = +1$, puis $u = -1$. On en déduit qu'il existe un unique couple (t_1, t_f) tel que

$$\begin{cases} x_1 = -\frac{t_f^2}{2} + t_1 t_f - t_1^2 \\ x_2 = -t_f + 2t_1 \end{cases}$$

La résolution de ce système conduit facilement à

$$t_f = -x_2 + \sqrt{-4x_1 + 2x_2^2} = -x_2 + 2\sqrt{-x_1 + \frac{1}{2}x_2^2}$$

De même, si le point (x_1, x_2) est au-dessus de la courbe γ , on calcule

$$t_f = x_2 + \sqrt{4x_1 + 2x_2^2} = x_2 + 2\sqrt{x_1 + \frac{1}{2}x_2^2}$$

Enfin, on a clairement $t_f = |x_2|$, finalement, la fonction temps minimal est donnée par la formule

$$t_f(x_1, x_2) = \begin{cases} -x_2 + 2\sqrt{-x_1 + \frac{1}{2}x_2^2} & \text{si } x_1 \geq \frac{x_2^2}{2}|x_2| \\ x_2 + 2\sqrt{x_1 + \frac{1}{2}x_2^2} & \text{si } x_1 \leq \frac{x_2^2}{2}|x_2| \end{cases} \quad (3.10)$$

On cherche par la suite le problème de contrôle en temps optimal avec la condition

$$\text{finale } \begin{pmatrix} x_{1f} \\ x_{2f} \end{pmatrix} = \begin{pmatrix} 0 \\ -1 \end{pmatrix}$$

D'après la formule (3.10), on a : $t_{f_{exact}} = 1 + \sqrt{2} \approx 2.4142$.

3.4 Application de la methode tir simple (Methode indirect) :

La méthode de tir simple permet de trouver la commande optimale et la trajectoire associée là où le PMP seul s'avère impuissant. C'est le cas de notre problème qui est de type Bang-bang.

Dans notre cas on résout le problème optimal en considérant l'état initial x_0 comme étant l'origine et en prenant un état final $x_1 = (0, -1)$ de l'espace X .

Il existe une fonction $z(t) = (x(t), p(t))$ définie dans X et à valeur dans U , et les conditions initiales, finales et les conditions de transversalité, se mettent sous

la forme :

$$R(z(0), z(t_f)) \quad (3.11)$$

et définit toutes les trajectoires optimales, et d'après le PMP on a $u(t) = \text{signe}(p_2(t))$. La connaissance de la fonction de tir permet de considérer que ce problème optimal est entièrement résolu mathématiquement.

Nous avons le problème aux deux bouts suivant :

$$(TBVP) = \begin{cases} \dot{x}_1(t) = x_2(t) \\ \dot{x}_2(t) = u(t) \\ \dot{p}_1(t) = 0 \\ \dot{p}_2(t) = -p_1(t) \\ x_1(t_0) = 0, \quad x_1(t_f) = 0 \\ x_2(t_0) = 0, \quad x_2(t_f) = -1 \end{cases} \quad (3.12)$$

En posons $z(t) = (x_1(t), x_2(t), p_1(t), p_2(t)) = (z_1(t), z_2(t), z_3(t), z_4(t))$.

Résoudre le problème $(TPBVP)$ est alors équivalent à rechercher un zéro de l'équation $F(z_0) = 0$ où la fonction F est la fonction de tir associée à notre problème et définie par :

$$F : \mathbb{R}^2 \longrightarrow \mathbb{R}^2 \quad (3.13)$$

$$z_0 \longrightarrow F(z_0) = R(t, 0, h) \quad (3.14)$$

avec $R(t, 0, h)$ est la solution du système suivant :

$$(TBVP) = \begin{cases} \dot{z}_1(t) = z_2(t) \\ \dot{z}_2(t) = u(t) \\ \dot{z}_3(t) = 0 \\ \dot{z}_4(t) = -z_3(t) \\ z_1(0) = 0, \quad z_3(0) = h_1 \\ z_2(0) = 0, \quad z_4(0) = h_2 \end{cases} \quad (3.15)$$

Soit $R(t, 0, 0, h_1, h_2)$ une solution du système au temps t avec les conditions initiales et finales $(0, 0, h_1, h_2)$.

Dans cet exemple on a : le temps final est libre c'est t_f , on doit avoir :

$$R(t_f, 0, 0, h_1, h_2) = \begin{pmatrix} R_1(t_f, 0, 0, h_1, h_2) \\ R_2(t_f, 0, 0, h_1, h_2) \\ R_3(t_f, 0, 0, h_1, h_2) \\ R_4(t_f, 0, 0, h_1, h_2) \end{pmatrix} = \begin{pmatrix} 0 \\ -1 \\ qlq \\ qlq \end{pmatrix}$$

On définit la fonction de tir suivante :

$$F(z_0) = \begin{pmatrix} R_1(t_f) \\ R_2(t_f) + 1 \\ H(t_f) = 0 \end{pmatrix}$$

qu'est égale à :

$$F(z_0) = \begin{pmatrix} R_1(t_f, 0, 0, h_1, h_2) - (0) \\ R_2(t_f, 0, 0, h_1, h_2) - (-1) \\ R_3(t_f, 0, 0, h_1, h_2)R_4(t_f, 0, 0, h_1, h_2) + |R_4(t_f, 0, 0, h_1, h_2)| - 1 \end{pmatrix}$$

car $(x_{1f}, x_{2f}) = (0, -1)$ et $\max H(t) = R_3(t)R_4(t) + |R_4(t)| - 1$, et le fait que le temps final t_f est libre donc $H(t_f) = R_3(t_f)R_4(t_f) + |R_4(t_f)| - 1 = 0$.

3.5 Résolution à l'aide du logiciel MATLAB :

On se ramène à un problème de tir simple, le programme suivant est effectué sous MATLAB, on utilise la méthode de newton, implémentée dans la Toolbox optim de MATLAB, il s'agit de la routine fsolve.m.

3.6 Méthode indirecte

3.6.1 fonction tirsimple

Code de la méthode de tir

```
function[r,tf,u]= methodeindirecte(x0,A,B,p0,tf,n,m,xf)
%methodeindirecte
%x0=[0 ;0];
%A=[0 1 ;0 0];
%B=[0 ;1];
%p0=[1 ;1];
%tf=5;
%on prend des valeurs initiales de p(0)et de tf.
%n=2;
%m=1;
%xf=[0 ;-1];
%n :dimention de l'espace X,m :dimention de l'espace U.
options=optimset('Display','iter','LargeScale','on');
[ptf,r]=fsolve(@(X)F(X,A,B,x0,xf,n,m),[p0 ;tf],options);
%on utilise fsolve.m pour trouver un zéro de F.
```

```

%dessin
tf=potf(end);
p0=potf(1 :end-1);
z0=[x0;p0];
options=odeset('AbsTol',1e-9,'RelTol',1e-9);
[Tz, z]=ode113(@(t,z)state(t,z,A,B,n,m),[0 tf],z0,options);
z=z';
x=z(1 :n, :);
p=z((n+1) :end, :);
q=B'*p;
u=zeros(m,length(Tz));
for i=1 :length(Tz)
u(:,i)=q(:,i)/norm(q(:,i),m);
end
subplot(121);
plot(x(1, :),x(2, :),'r');
axis square;
title('Trajectoire');
subplot(122);
plot(Tz,u,'r');
axis square;
title('Contrôle');
%
r
%
tf
%
u
function res=F(X,A,B,x0,xf,n,m)
%F rend la distance entre x(tf)et xf
tf=X(end);
p0=X(1 :(end-1));
z0=[x0;p0];
options=odeset('AbsTol',1e-9,'RelTol',1e-9);
[Tz, z]=ode113(@(t,z)state(t,z,A,B,n,m),[0 ;tf],z0,options);
% z représente les valeurs de x en chaque instant.on utilise ode113 pour
% résoudre le pb de cauchy
res=z(end,1 :n)'-xf;
end

```

```

function r=state(t,z,A,B,n,m)
x=z(1 :n) ;
p=z((n+1) :end) ;
q=B'*p ;
u=q/norm(q,m) ;
%on rappelle que u(t)=b*p(t)/|B*p(t)|si il est en temps optimal
r=[A*x+B*u ;-A'*p] ;
end

```

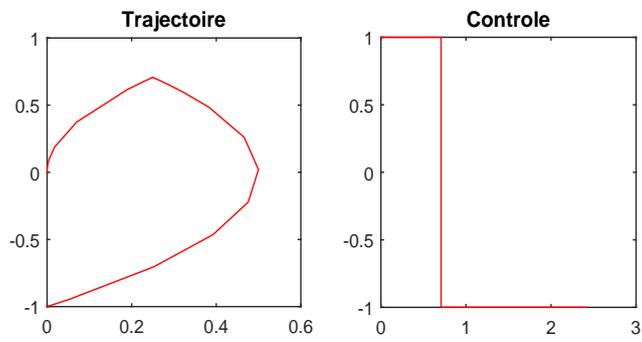


FIGURE 3.3 – $p0_{init} = [1; 1], t_{finit} = 5$

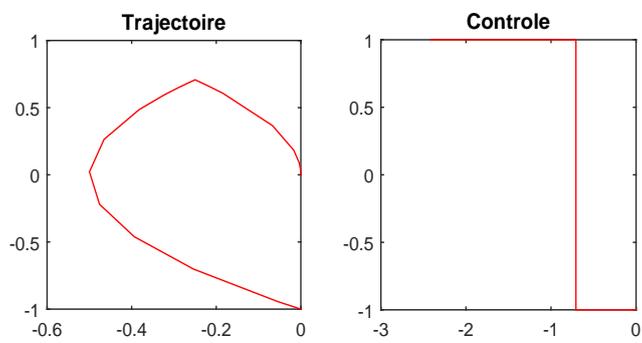


FIGURE 3.4 – $p0_{init} = [1; 1], t_{finit} = 100$

Conditions initiales	t_f
$p_{0init} = [1; 1], t_{finit} = 5$	2.4142
$p_{0init} = [1; 1], t_{finit} = 100$	-2.4142

TABLE 3.1 – Cas 1 : temps final numérique

3.7 Application de la discrétisation totale (Méthode directe) :

3.7.1 Résolution à l'aide du logiciel MATLAB :

function direct

```
_____ Code de la méthode de discretisation totale _____  
    clc ;clear all ;close all  
N=input('donner N=...') ;%exemple 100  
A=input('donner A=...') ;%exemple [0 1 ;0 0]  
B=input('donner B=...') ;%exemple [0 ;1]  
n=input('donner n=...') ;%exemple 2  
m=input('donner m=...') ;%exemple 1  
tf=input('donner tf=...') ;%exemple 5  
x0=input('donner x0=...') ;%exemple zeros(n,1)  
xf=input('donner xf=...') ;%exemple [0 ;-1]  
function[tf,U]=methodedirecte(N,A,B,n,m,tf,x0,xf)  
U=2*rand(m,N)-1 ;  
X0=[reshape(U,N*m,1) ;tf] ;  
options=optimset('Algorithm','sqp') ;  
%on choisit la méthode sqp comme méthode d'optimisation non lineaire.  
function[X,tf]=fmincon(@principal,X0,[],[],[],[],[],@(X)contrainte(X,A,B,m,N,x0,xf),o  
%dessins  
U=X(1 :(end-1)) ;  
U=reshape(U,m,N) ;  
x=zeros(n,N+1) ;  
h=tf/N ;  
for i=1 :N  
x( :,i+1)=x( :,i)+h*(A*x( :,i)+B*U( :,i)) ;  
end  
subplot(121) ;  
plot(x(1, :),x(2, :),'r') ;  
axis square ;  
title('Trajectoire') ;  
subplot(122) ;  
plot(linspace(0,tf,N),U,'r') ;  
axis square ;  
title('contrôle') ;  
end
```

```

function [c,ceq]=contrainte(X,A,B,m,N,x0,xf)
tf=X(end);
U=X(1 :(end-1));
U=reshape(U,m,N);
h=tf/N;
c=[sum(abs(U).^m,1).^(1/m) - 1, -tf];
x=x0;
%c représente la contrainte qu'en chaque instant ti,|U(ti)|<=1 et tf>=0
for i=1 :N
x=x+h*(A*x+B*U( :,i));
%on utilise le schéma d'euler explicite pour résoudre le système.
end
ceq=x-xf;
function t = principal(X)
t=X(end);
end

```

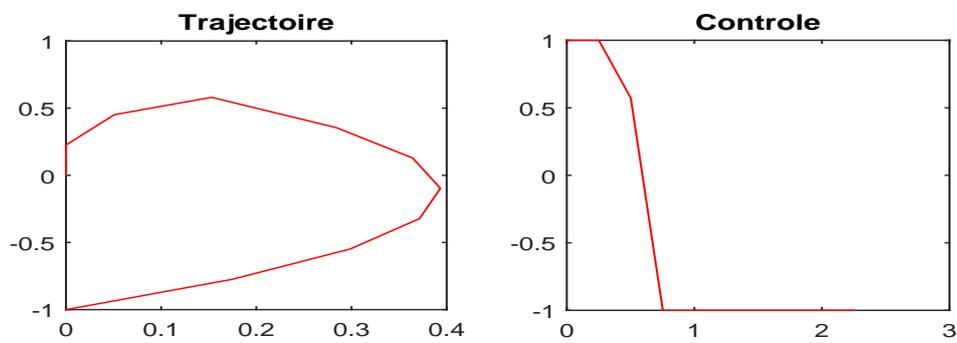


FIGURE 3.5 – Les grands croix pour $N = 10$

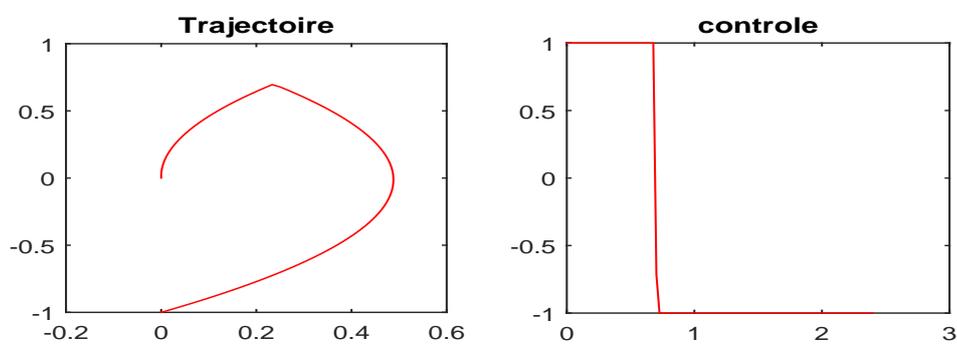


FIGURE 3.6 – Les grands croix pour $N = 100$

N	$_{f}f$
10	2.2581
100	2.3973

TABLE 3.2 – Cas 2 : temps final numérique

Remarque

Si on remarque que pour la méthode directe, si on augmente le nombre de discrétisation, le temps final numérique est plus proche du temps final exact. De plus, le résultat obtenu par la méthode indirect est très précis par rapport la méthode direct.

Conclusion Générale

Notre objectif dans ce memoire est l'étude d'un problème de contrôle optimal . dans le chapitre premiere on a présenter quelques concepts de bases de la théorie de contrôle optimal et quelques définitions de bases et les différents systèmes de contrôle optimal

Le problème de Contrôle optimal, peut être interpréter comme un problème pour lequel on cherche un meilleur départ appartenant à un ensemble redéfini X_0 et une commande u qui nous permet de ramener le système de l'état initial $x(0)$ à l'état final $x(t_f)$.

Dans le deuxième chapitre nous avons expliqué comment résoudre le problème de contrôle optimal. Où il a été étudié deux grandes classes de méthodes : les méthodes directes et méthodes indirectes.

- Les méthodes directes consistent à discrétiser le problème de contrôle pour tout instant puis le résoudre en se basant par exemple sur les méthodes d'Euler, Runge Kutta..., ces méthodes sont les plus simple à mettre en oeuvre.
- les méthodes indirectes quant à elles, consistent à appliquer le principe du maximum de pontiyagin qui donne les conditions nécessaire d'optimalité du 1^{er} ordre.

Il serait préférable d'opter pour une méthode indirecte :

- Si la dimension d'espace est assez grande.
- Si on a besoin de calculer la trajectoire optimale d'une manière très précise.
- Dans un deuxième temps, après avoir appliqué une méthode directe qui a donné une première approximation de la solution optimale.

En fin on a donné un exemple qu'on a résolu sous Matlab en utilisant la méthode de tir simple et la méthode de tir directe .

Annexe A

Méthode d'Euler explicite

Définition A.1. Soit une équation différentielle pour la fonction y de la variable t de la forme suivante :

$$\frac{dy}{dt} = f(t, y(t))$$

avec la condition initiale

$$y(0) = y_0$$

La variable t sera appelée le temps. Dans le cas d'un système différentiel du premier ordre, y désigne une matrice colonne comportant les fonctions inconnues.

L'intégration numérique d'une équation de ce type consiste à calculer des valeurs approchées de y sur un intervalle $[0, T]$. Pour cela, on divise cet intervalle en N sous-intervalles de longueur $h=T/N$ et on définit les instants :

$$t_n = nh$$

où l'entier n varie de 0 à N . Dans la méthode d'Euler explicite, la valeur approchée à l'instant t_{n+1} est obtenue à partir de la précédente par

$$y_{n+1} = y_n + hf(t_n, y_n)$$

On définit l'erreur (globale) à l'instant t_n par :

$$e_n = |y(t_n) - y_n|$$

Bien entendu, la résolution numérique n'a d'intérêt que si la solution exacte $y(t)$ ne peut être déterminée. Aussi l'erreur ne peut être connue avec précision mais nous verrons qu'il est possible de savoir comment cette erreur dépend du pas h . Une autre question très importante est l'évolution de l'erreur e_n en fonction de n : c'est le problème de la stabilité.

Supposons que $y(t_n)$ soit connu sans imprécision. On appelle erreur locale l'erreur

introduite pour le calcul à l'instant suivant. La méthode d'Euler repose sur le développement de Taylor suivant :

$$y(t_{n+1}) = y(t_n) + h \frac{d_y}{d_t}(t_n)$$

dans lequel le terme négligé est d'ordre h^2 . Pour cette raison, la méthode d'Euler est dite d'ordre 1. L'erreur locale est donc vraisemblablement proportionnelle à h^2 , ce qui signifie qu'une réduction du pas h d'un facteur 2 réduit l'erreur locale d'un facteur 4[5, 10].

Annexe B

Méthode de Runge Kutta

Les techniques de Runge-Kutta sont des schémas numériques à un pas qui permettent de résoudre les équations différentielles ordinaires. Elles font parties des méthodes les plus populaires de part leur facilité de mise en œuvre et leur précision. C'est Carle Runge et Martin Kutta qui, au début du XXe siècle, ont inventé ces méthodes. Nous décrivons ici deux algorithmes assez utilisés : celles de Runge-Kutta d'ordre 2 et 4.

Dans de nombreux cas, les systèmes d'équations différentielles que l'on rencontre en science peuvent se mettre sous la forme d'une équation différentielle ordinaire du premier ordre du type :

$$\begin{cases} \frac{dy}{dt} = f(t, y(t)), & 0 \leq t \leq T \\ y(0) = y_0 \end{cases}$$

où $y(t)$ est la fonction que l'on recherche, y_0 sa valeur initiale et f une fonction connue suffisamment régulière pour que l'existence et l'unicité de la solution ne pose pas de problème. Notez que $y(t)$ peut être un scalaire ou un vecteur[10]

Annexe C

Théorie des points conjugués

Le but de la théorie des points conjugués est de donner des conditions (du second ordre) nécessaires et/ou suffisantes d'optimalité, pour des problèmes de contrôle optimal. Dans le cas général, on se réfère à [25]. Dans cette annexe, pour simplifier la présentation, on se limite au problème du temps minimal pour le système de contrôle

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

où $f : R^n \times R^m \rightarrow R^n$ est une application lisse, $x_0 \in R^n$, et $u(t) \in R^m$ (cas sans contrainte sur le contrôle). Tout contrôle u temps minimal sur $[t_0, t_f]$ est alors singulier. D'après le principe du maximum qu'est énoncé au chapitre précédent, la trajectoire $x(\cdot)$ est projection d'une extrémale, solution des équations

$$\dot{x} = \frac{dH}{dP}(x, p, u), \quad \dot{p} = -\frac{dH}{dx}(x, p, u) \quad \text{et} \quad \frac{dH}{du}(x, p, u) = 0,$$

où $H(x, p, u) = \langle p, f(x, u) \rangle$. On suppose que $H(x(t), p(t), u(t)) \neq 0$ le long de l'extrémale, que la Hessienne $\frac{d^2H}{du^2}(x, p, u)$ est définie négative (hypothèse de Legendre stricte), et que la singularité est de codimension un sur tout sous-intervalle (hypothèse de régularité forte). L'application du théorème des fonctions implicites permet d'exprimer, localement, $u(t) = u(x(t), p(t))$. Toute extrémale vérifie alors

$$\dot{x} = \frac{dH_r}{dP}(x, p), \quad \dot{p} = -\frac{dH_r}{dx}(x, p),$$

où $H_r(x, p) = H(x, p, u(x, p))$ est le Hamiltonien réduit. Et notant $z = (x, p)$,

$$z(t) = H_r^{\rightarrow}(z(t)). \quad (M.1)$$

On définit l'application exponentielle

$$\exp x_0(t, p_0) = x(t, x_0, p_0),$$

où $(x(t, x_0, p_0), p(t, x_0, p_0))$ est la solution du système (M.1) partant du point (x_0, p_0) en $t = 0$. Le temps t varie dans R^+ (du moins, tant que la solution est bien définie), et le vecteur adjoint initial p_0 est défini à scalaire multiplicatif près, sachant que l'on suppose $H_r(x_0, p_0) \neq 0$, on peut donc supposer que $p_0 \in S_{n-1}$. finalement, l'application exp_{x_0} est définie sur $R^+ \times S_{n-1} \cap p_0 | H_r(x_0, p_0) \neq 0$, à valeurs dans R^n . Son image s'appelle un champ central. Soit $z(t, z_0) = (x(t, x_0, p_0), p(t, x_0, p_0))$ une extrémale de référence, définie sur $[t_0, t_f]$. Le temps $t_c \in]t_0, t_f]$ est dit conjugué si l'application $exp_{x_0}(t_c, \cdot)$ n'est pas une immersion en p_0 . Le point $x(t_c)$ est dit conjugué à x_0 .

Pour calculer un temps conjugué, il s'agit donc de déterminer un temps auquel le rang de $\frac{dexp_{x_0}(t_c, \cdot)}{dp_0}$ est inférieur ou égal à $n - 2$. Pour cela, on introduit le système variationnel

$$\delta \dot{z}(t) = \frac{dH_r^\rightarrow}{dz}(z(t)) \delta z(t), \quad (M.2)$$

ou appelé équation de Jacobi. Un champ de Jacobi $J(t) = (\delta x(t), \delta p(t))$ est une solution non triviale de (M.2). Il est dit vertical au temps t si $\delta x(t) = 0$. Un temps t_c est conjugué si et seulement s'il existe un champ de Jacobi vertical aux temps 0 et t_c . La justification de ce résultat technique se trouve dans [25].

Annexe D

La programmation quadratique séquentielle (SQP)

La programmation quadratique séquentielle (SQP) est utilisée pour la résolution des problèmes généraux d'optimisation non linéaire avec contraintes. Les problèmes à résoudre avec cette méthode sont de la forme :

$$\begin{cases} \min_{x \in R^n} f(x) \\ \text{sous } g_i(x) = 0 \text{ pour } i = 1, \dots, n \\ h_j(x) \geq 0, \text{ pour } i = n + 1, \dots, m \\ x^{inf} \leq x \leq x^{sup} \end{cases}$$

Où toutes les fonctions du problème sont continues différentiables. La programmation quadratique séquentielle basée sur la formulation itérative et la solution de sous problème de la programmation quadratique, on obtient le sous problème en utilisant une approximation quadratique de Lagrangien et par linéarisation des contraintes [19].

Les avantages :

avec la programmation quadratique on peut résoudre tous les types d'optimisation non linéaire avec contraintes (quadratique ou non quadratique), et quelque soit le type de contraintes (contraintes d'égalité ou contraintes d'inégalité).

- Cette méthode est bénéficiée d'une convergence quadratique.
- Cette méthode utilise une recherche linéaire pour calculer un pas d'optimisation.

Annexe E

Conditions de Lagrange

Théorème E.1. Soit $x^* \in S = \{x \in R^n, g(x) = 0\}$ un point régulier solution du problème d'optimisation, vérifiant donc : $f(x^*) \leq f(x), \forall x \in S$.

Et Soit la matrice Jacobienne $J(x)$ de dimension $m \times n$:

$$J(x) = \begin{bmatrix} \nabla g_1(X)^T \\ \cdot \\ \cdot \\ \cdot \\ \nabla g_m(X)^T \end{bmatrix} \iff J^T(x) = \nabla g^T(x)$$

Un point x^* tel que $g(x^*) = 0$ est appelé un point régulier des contraintes si $\text{rang } j(x^*) = m$ (Voir [20])

Annexe F

Méthodes de pénalité

Les méthodes de pénalité cherchent à approcher le minimum depuis l'extérieur de X . A la fonction $f(x)$ est ajoutée une fonction de pénalité extérieure $P(x)$ dont la valeur est égale à zéro si x est admissible et supérieure à zéro s'il ne l'est pas. L'ajout de cette fonction a pour seul but de pénaliser la fonction objectif en cas de violation d'une ou de plusieurs contraintes [21]. La fonction de pénalité quadratique $P(x)$ est définie comme suit :

$$P(x) = \sum_{i=1}^m (h_i(x))^2 u_i(h_i(x))$$

Où

$$u_i(h_i(x)) = \begin{cases} 0 & \text{si } h_i(x) \leq 0 \\ 1 & \text{si } h_i(x) > 0 \end{cases}$$

Annexe G

Types d'équations différentielles

G.0.1 Équation différentielle à variables séparées

Une équation différentielle est dite à variables séparées si elle peut s'écrire sous la forme :

$$\dot{x} = g(x)f(t)$$

Pour la résoudre, on intègre les deux membres séparément, sans oublier les constantes d'intégration.

$$\dot{x} = \frac{dx}{dt} = g(x)f(t) \Rightarrow \frac{dx}{g(x)} = f(t)dt$$

par intégration, on obtient

$$\int \frac{dx}{g(x)} = \int f(t)dt$$

Soit l'équation

$$\frac{dx}{dt} = -\frac{x}{t}$$

Séparons les variables :

$$\frac{dx}{x} = -\frac{dt}{t}$$

On trouve par intégration :

$$\int \frac{dx}{x} = -\int \frac{dt}{t} + C$$

donc

$$\text{Log}|x| = -\text{Log}|t| + \text{Log}|C| \Rightarrow \text{Log}|x| = \text{Log}\left|\frac{C}{t}\right|$$

d'où l'en déduit la solution générale $x = \frac{C}{t}$.

G.0.2 équation différentielle linéaire d'ordre 1

Une équation différentielle est dite linéaire si elle peut s'écrire sous la forme :

$$\frac{dx}{dt} + P(t)x = Q(t) \quad (\text{G.1})$$

où $P(t)$ et $Q(t)$ sont des fonction continue de t données .

- Si $Q(t) = 0$ alors (G.1) est dite homogène.
- Si $Q(t) \neq 0$ alors (G.1) est dite non homogène.

Les solutions de l'équation (G.1) si $Q(t) = 0$ est la solution de l'équation à variables séparées suivante :

$$\frac{dx}{dt} = -P(t)x \iff \frac{dx}{x} = -P(t)dt$$

l'intégrale générale donné par :

$$x = C e^{-\int P(t)dt} \quad (\text{G.2})$$

Les solutions de l'équation (G.2) s'obtiennent en ajoutant une solution particulière de cette équation à la solution générale de l'équation homogène associée.

Annexe H

méthode de Galerkin

La méthode de Galerkin est une méthode très générale et très robuste. L'idée de la méthode est la suivante. Partant d'un problème posé dans un espace de dimension infinie, on procède d'abord à une approximation dans une suite croissante de sous-espaces de dimension finie. On résout ensuite le problème approché, ce qui est en général plus facile que de résoudre directement en dimension infinie. Enfin, on passe d'une fac on ou d'une autre à la limite quand on fait tendre la dimension des espaces d'approximation vers l'infini pour construire une solution du problème de d epart. Il convient de noter que, outre son interet theorique , la méthode de Galerkin fournit egalement un procédé constructif d'approximation.(pour plus sur la méthode voir [\[32\]](#))

Bibliographie

- [1] E.Trealat, Contrôle optimal : Théorie et Applications. université Paris-Sud, laboratoire EN- EDP mathématiques UMR 8628 Bat u25, 91405 osay cedex 2005.
- [2] A. V. Rao. A survey of numerical methods for optimal control. Applied Mathematics and Computation, 345 :543–548, 2007.
- [3] R. W. H. Sargent. Optimal control. Journal of Computational and Applied Mathematics, 124 :361-371, 2000.
- [4] S. Boyd and L. Vandenberghe. Convex Optimization. Cambridge, United Kingdom : Cambrige University Press, 2004.
- [5] R. Bellman. Dynamic Programming. Princeton University Press, Princeton, New Jersey, 1957.
- [6] R. W. Beard, G. Saridis, and J. T.Wen. Galerkin approximations of the generalized Hamilton- Jacobi-Bellman equation. Automatica, 33 :2159–2177, 1997.
- [7] F. Bauer, L. Grune, and W. Semmler. Adaptive spline interpolation for Hamilton-Jacobi- Bellman equations. Applied Numerical Mathematics, 56 :1196–1210, 2006.
- [8] H. B. Keller. Numerical Solution of Two Point Boundary Value Problems. SIAM, 1976
- [9] J. Stoer, R. Bulirsh, Introduction to Numerical Analysis , second edition 1991
- [10] A. V. Rao. A survey of numerical methods for optimal control. Applied Mathematics and Computation, 345 :543–548, 2007.
- [11] J. I. Ramos. On the variational iteration method and other iterative techniques for nonlinear differential equations. Applied Mathematics and Computation, 199(1) :39–69, 2008.
- [12] G. Adomian. Stochastic Systems. Academic Press, Inc., New York, 1983.
- [13] J. H. He. Variational iteration iteration method- a kind of nonlinear analytical technique : Some examples. International Journal of Non-Linear Mechanics, 34 :699-708, 1999

- [14] J. H. He. Asymptotology by homotopy perturbation method. Applied Mathematics and Com- putation, 156 :591-596, 2004.
- [15] A. Fakharian and M. T. H. Beheshti. Solving linear and nonlinear optimal control problem using modified adomian decomposition method. Journal of Computer and Robotics, 1, 1–8, 2008.
- [16] F. Ghomanjani, S. Ghaderi, and M. H. Farahi. Solving the optimal control of linear systems via homotopy perturbation method. Intelligent Control and Automation, 3 :26–33, 2012.
- [17] D. Jacobson, D. Lele, J. L. Speyer, New necessary conditions of optimality for control problems with state-variable inequality constraints, Journal of Mathematical Analysis and Applications, Vol. 35, pp. 255–284, 1971.
- [18] R. Fletcher, Practical Methods of Optimization, Vol. 1, Unconstrained Op- timization, and Vol. 2, Constrained Optimization, John Wiley and Sons, 1980.
- [19] IMSL, Fortran Subroutines for Mathematical Applications, IMSL STAT/LIBRARY User’s Manual, Version 2.0, IMSL, Houston, 1991.
- [20] S. Mottelet, Optimisation Non-linéaire -RO04/TI07, Université de Technologie de Compiègne, Printemps 2003
- [21] T. Diallo, Etude et Illustration de Méthodes Itératives D’optimisation non Linéaire, Projet de master 2005-2006, Ecole Polytechnique Fédérale de Lausanne, Faculté Sciences de Base, Chaire de Recherche Opérationnelle Sud Est, Lausanne, avril 2006
- [22] E. B. Lee, L. Markus, Foundations of optimal control theory, John Wiley, New York, 1967.
- [23] C. Vachier, Représentation d’état et commande dans l’espace d’état, notes de cours, UNIVERSITÉ PARIS XII-VAL DE MARNE, 2007.
- [24] C. Lobry et T .Sari, Tntroduction à la théorie du contrôle. CIMPA, Tlemcen, 2003.
- [25] B. Bonnard, J-B. Caillau et E. Tr éelat, Second order optimality conditions and applications in optimal control. Preprint de l’Univ. d’Orsay, 2005.
- [26] R.E. Kalman. Contributions to the theory of optimal control. Bolet éin de la Sociedad Matem éatica Mexicana, 5(2) :102–119, 1960.
- [27] Nenik Nijmejer, Arjan Van Der Shaft, Nonlinear Dynamical Control Systems. Springer-Verlag New Yourk Inc. 1990.
- [28] B. Bonnard et M. Chyba, The role of singular trajectories in control theory. Math. Monograph, Springer-Verlag, 2003.
- [29] G.B. Dantzig, Programmation lin éeaire. Edition : Dunod, Paris, 1966.

- [30] F. Boyer. A propos de contrôle et de l'optimisation de trajectoires. Laboratoire d'Analyse Topologie et Probabilités, Aix Marseille University. Aix en Provence., Decembre 2010.
- [31] J. A. Sethian, Level set methods and fast marching methods. Evolving interfaces in computational geometry, fluid mechanics, computer vision, and materials science. Cambridge Monographs on Applied and Computational Mathematics, 3, Cambridge University Press, 1999.
- [32] Curnier A. : Méthodes Numériques en Mécanique des Solides, Presses Polytechniques et Universitaires Romandes, Lausanne, ISBN 2-88074-247-1, 1993