

Table des matières

Notations	3
Introduction générale	4
1 Équation Différentielle Ordinaire	6
1.1 Introduction :	6
1.2 Définitions	6
1.3 Solutions maximales et globales	6
1.4 Réduction de l'ordre	7
1.5 Existence et unicité :	7
1.5.1 Problème de Cauchy :	7
1.6 Quelques techniques de résolution	9
1.6.1 Équations linéaires du 1 ^{er} ordre	9
1.6.2 Équations à variables séparables	9
1.6.3 Équations de Bernoulli	9
1.6.4 Équations de Riccati :	10
1.7 Méthodes numérique de résolution des équations différentielles	10
1.7.1 Formulation générale :	10
1.7.2 Méthode d'Euler	11
1.7.3 Méthode de Runge-Kutta d'ordre 2	12
1.7.4 Méthode de Runge-Kutta d'ordre 4	12
1.8 Conclusion	13
2 Théorie du Contrôle Optimale	14
2.1 Introduction	14
2.2 Définitions :	14
2.3 Positions du problème	15
2.3.1 Problème de Lagrange :	16
2.3.2 Problème de Mayer :	16
2.3.3 Classement des problèmes de contrôle optimal	16
2.4 Contrôlabilité :	17
2.4.1 Notion de contrôlabilité	17
2.4.2 Ensemble d'accessibilité	17
2.4.3 Contrôlabilité des systèmes linéaires :	18
2.4.4 Contrôlabilité des systèmes linéaires autonome :	19
2.4.5 Contrôlabilité des systèmes non-linéaires	19
2.5 Principe du maximum :	20
2.5.1 Principe du maximum faible :	20
2.5.2 Principe de maximum de Pontryagin (PMP) :	22
2.5.3 Condition de transversalité :	22
2.5.4 Exemple :	22

2.6	Méthodes numériques en contrôle optimal :	24
2.6.1	Méthodes directes	24
2.6.2	Méthodes indirectes	26
2.6.3	Quelle méthode choisir ?	29
2.7	Exemple :	29
2.7.1	Résolution par la méthode indirecte sous MatLab :	30
2.7.2	Résolution par la méthode directe sous MatLab :	32
2.8	Conclusion	34
3	Contrôle optimal du problème de GODDARD	35
3.1	Introduction	35
3.2	Généralités :	35
3.2.1	La fusée :	35
3.2.2	Les phases de vol :	35
3.2.3	Les forces en présence :	36
3.3	Rappels sur les lois de la physique (Newton)	37
3.3.1	La première loi de Newton	38
3.3.2	La deuxième loi de Newton	38
3.4	Position du problème :	38
3.4.1	Résolution théorique (Principe du maximum)	41
3.4.2	Résolution numérique	45
3.5	Discussion des résultats	50
	Conclusion générale	52
	Bibliographie	53

Notations

\mathbb{R} : ensemble des nombres réels

\mathbb{R}^n : $\mathbb{R} * \mathbb{R} * \mathbb{R}$ n fois

\mathbb{R}^m : $\mathbb{R} * \mathbb{R} * \mathbb{R}$ m fois

C^1 : classe 1

$\dot{x} = \frac{dx}{dt}$ dérivée de x par rapport à t

$||$ valeur absolue

$Acc(x_0, T)$: ensemble d'accessibilité depuis x_0 , en temps T

\forall : quelque soit

\exists : il existe

$\partial Acc(x_0, T)$: la frontière de l'ensemble accessibilité

$M_n(\mathbb{R})$: ensemble des vecteurs à n lignes à coefficients dans \mathbb{R}

$M_{n,m}(\mathbb{R})$: ensemble des matrices à n lignes et à m colonnes à coefficients dans \mathbb{R}

id : identité

e^A : exponentielle de la matrice A

min : minimum

max : maximum

$h(t)$: l'altitude au temps t .

$v(t)$: le module de la vitesse.

$m(t)$: la masse de la fusée au temps t .

$d(h(t), v(t))$: la densité de l'air

$q(h(t), v(t))$: la pression atmosphérique

β : taux de décroissance de la densité de l'air

c : vitesse du combustion du carburant

p_0 : densité de l'air au niveau de la mer

C_d : coefficient de la pression atmosphérique

A : surface frontale de contact

g : la gravité terrestre

M_0 : la masse initiale

Introduction générale

La théorie du contrôle analyse les propriétés des systèmes commandés, c'est-à-dire des systèmes dynamiques sur lesquels on peut agir au moyen d'une commande (ou contrôle). Le but est alors d'amener le système d'un état initial donné à un certain état final, en respectant éventuellement certains critères. Les systèmes abordés sont multiples : systèmes différentiels, systèmes discrets, systèmes avec bruit, avec retard...Leurs origines sont très diverses : mécanique, électricité, électronique, biologie, chimie, économie...L'objectif peut être de stabiliser le système pour le rendre insensible à certaines perturbations (stabilisation), ou encore de déterminer des solutions optimales pour un certains critères d'optimisation (contrôle optimal). Le point clé de cette théorie est le principe du maximum de Pontryagin, qui donne une condition nécessaire d'optimalité et permet ainsi de calculer les trajectoires optimales.

On considère que la théorie moderne du contrôle optimal a commencée dans les années cinquante, avec la formulation du principe du maximum de Pontryagin, qui généralise les équations d'Euler-Lagrange du calcul des variations. Dès lors, la théorie a connu un essor spectaculaire, ainsi que de nombreuses applications. De nos jours, les systèmes automatisés font complètement partie de notre quotidien (nous en sommes souvent inconscients), ayant pour but d'améliorer notre qualité de vie et de faciliter certaines tâches : systèmes de freinage, assistance à la conduite, servomoteur, thermostats, circuits frigorifiques, contrôle des flux routiers, circuits électriques, électroniques, télécommunications en général, contrôle des procédés chimiques, chaînes industrielles, ...La liste est infinie, les applications concernent tout système sur lequel on peut avoir une action, avec une notion de rendement optimal.

Le contrôle optimal sert à trouver une lois de commande pour un système donné de telle sorte qu'un certain critère d'optimalité soit atteint. Un problème de commande comprend donc un coût à optimiser, une fonction d'état est une variable de contrôle. Un problème de contrôle optimal se décompose en deux parties : pour déterminer une trajectoire optimale joignant un ensemble initial à une cible, il faut d'abord savoir si cette cible est atteignable, c'est le problème de contrôlabilité. Pour les systèmes de contrôle linéaire en dimension finie, il existe une caractérisation très simple de la contrôlabilité, apparue dans les années soixante avec les travaux de R.E.Kalman, et pour les systèmes non linéaires, le problème de contrôlabilité est beaucoup plus difficile.

Une fois le problème de contrôlabilité est résolu, il faut chercher parmi toutes les trajectoires possibles celle qui donne le coût minimum (maximum) et pour ce faire, on dispose de deux grandes classes de méthodes à savoir la méthode directe et la méthode indirecte.

Le plan de notre manuscrit est le suivant :

Le premier chapitre sera consacré aux notions de résolution numérique des équations différentielles qui sont la base fondamentale du contrôle optimal. Nous avons défini quelques méthodes de résolution des équations différentielles du premier ordre les plus utilisées telles que la méthode d'Euler et celle de Runge-Kutta.

dans le second chapitre, on s'intéresse plus a la théorie du contrôle optimale, suivie des méthodes numériques de résolution d'un problème de contrôle optimale.

Enfin le chapitre trois est consacré a la résolution numérique du problème de GODDARD qui consiste a minimiser l'énergie consommée pendant le vol d'une fusée, Et nous terminons le manuscrit par une conclusion générale.

Chapitre 1

Équation Différentielle Ordinaire

1.1 Introduction :

Les équations différentielles jouent un rôle important dans différents domaines, en physique, en chimie, en économie, Ces équations ont rarement des solutions exactes, autrement dit, la résolution explicite de la plupart des équations différentielles ordinaires reste encore un problème ouvert. Ces équations ne s'intègrent pas d'une manière exacte. On fait appelle dans ce cas à des méthodes numériques afin de trouver des solutions approchées.

1.2 Définitions

Une équation différentielle ordinaire d'ordre n est une relation entre la variable réelle t , une fonction inconnue $t \rightarrow y(t)$ et ses dérivées y', y'', \dots, y^n au point t définie par :

$$f(t, y, y', y'', \dots, y^n) = 0 \quad (1.1)$$

Où f n'est pas indépendante de sa dernière variable y^n . On prendra t dans un intervalle I de \mathbb{R} (I peut être \mathbb{R} tout entier).

Le qualificatif ordinaire pour l'équation (1.1) signifie que la fonction inconnue dépend d'une variable t . lorsqu'il y a plusieurs variables t et plusieurs dérivées on parle d'équations aux dérivées partielles (EDP).

On appelle équation différentielle normale d'ordre n toute équation de la forme

$$y^n = f(t, y, y', y'', \dots, y^{(n-1)}) \quad (1.2)$$

Si l'équation (1.2) ne dépend pas de t on l'appelle équation différentielle autonome.

1.3 Solutions maximales et globales

Soient (x, I) et (\tilde{x}, \tilde{I}) deux solutions d'une même équation différentielle. On dira que (\tilde{x}, \tilde{I}) est un prolongement de (x, I) si $I \subset \tilde{I}$ et $\tilde{x}|_I = x$.

Solution maximale

Soient I_1 et I_2 , deux intervalles de \mathbb{R} tel que $I_1 \subset I_2$.

On dit qu'une solution (x, I_2) est maximale dans I_2 si et seulement si x n'admet pas de prolongement (\tilde{x}, \tilde{I}) solution de l'équation différentielle telle que $I_1 \subsetneq \tilde{I} \subset I_2$.

Solution globale

Soit I un intervalle inclus dans \mathbb{R} . Une solution (x, I) est dite globale dans I si elle est définie sur l'intervalle I tout entier.

En reprenant les même notations que dans les définitions précédentes, si une solution (x, I_1) peut se prolonger sur l'intervalle I_2 tout entier, alors x est globale dans I_2 .

Toute solution globale sur un intervalle I est maximale sur I ; mais la réciproque est fausse.

1.4 Réduction de l'ordre

Toute équation différentielle d'ordre n normalisée peut s'écrire sous la forme d'un système différentielle d'ordre 1, en effet posons :

$$y = y_1, \quad y' = y_2, \quad y'' = y_3, \dots, \quad y^{(n-1)} = y_n$$

L'équation $y^{(n)} = f(t, y, y', y'', \dots, y^{(n-1)})$ est équivalente au système différentiel d'ordre 1 suivant :

$$y_1' = y_2$$

$$y_2' = y_3$$

$$y_3' = y_4$$

$$\dots$$

$$y_n' = f(t, y, y', y'', \dots, y^{(n-1)})$$

1.5 Existence et unicité :

1.5.1 Problème de Cauchy :

Soit I_0 un intervalle de \mathbb{R} non réduit à un point et t un point fixé dans I_0 . On se donne une fonction f définie et continue sur $I_0 \times \mathbb{R}^m$ à valeur dans \mathbb{R}^m , ainsi qu'un élément y_0 dans \mathbb{R}^m et on cherche à trouver une fonction y continue et dérivable sur l'intervalle I_0 à valeur dans \mathbb{R}^m , telle que

$$\forall t \in I_0, \quad y'(t) = f(t, y(t)) \quad (1.3)$$

$$y(t_0) = y_0 \quad (1.4)$$

Ce problème s'appelle problème de Cauchy pour le système différentiel (1.3) et la condition (1.4) s'appelle la condition initiale. Une fonction y vérifiant les équations (1.3)-(1.4) est appelée une intégrale du système différentiel (1.3).

Théorème d'existence :

On suppose que la fonction f est continue dans un voisinage du point (t_0, y_0) dans $I_0 \times \mathbb{R}^m$ alors il existe un intervalle $J_0 \subset I_0$; au voisinage de t_0 et une fonction $y \in C^1(J_0)$ tels que

$$\forall t \in J_0, \quad y'(t) = f(t, y(t)) \quad (1.5)$$

Théorème d'unicité :

On dira que le problème (1.3)-(1.4) admet une et une seule solution, s'il admet une solution globale et si toute solution locale est la restriction de cette solution globale .

Théorème 1.5.1 (Cauchy Lipschitz) [1]

Considérons le problème (1.3)-(1.4)

avec $f : (t, y) \in I_0 \times \mathbb{R}^m \rightarrow f(t, y) \in \mathbb{R}^m$

- f est continue sur $I_0 \times \mathbb{R}^m$

- f est lipschitzienne en y , localement en y c'est à dire : il existe un réel $L > 0$ tel que

$\forall t \in I_0$ et $\forall y_1, y_2 \in v_{\mathbb{R}^m}(y_0)$

$$\| f(t, y_1) - f(t, y_2) \| \leq L \| y_1 - y_2 \| \quad (1.6)$$

alors le problème (1.3)-(1.4) admet une et une seule solution. Cette solution est définie sur un intervalle contenant t .

1.6 Quelques techniques de résolution

Dans cette section, nous allons nous intéresser à différentes techniques pour intégrer (c'est-à-dire résoudre), certains types d'équations différentielles. Il faut cependant garder à l'esprit que la résolution explicite des EDO n'est pas une chose aisée, et la plupart du temps ce sera trop difficile. Nous devons alors nous contenter d'une analyse d'existence, unicité, positivité, ...etc des solutions.

1.6.1 Équations linéaires du 1^{er} ordre

Une équation différentielle est dite linéaire du premier ordre si et seulement si elle est de la forme :

$$a(t)y' + b(t)y = c(t) \quad (1.7)$$

Où $a(t)$, $b(t)$ sont les coefficients et $c(t)$ le second membre de l'équation différentielle.

On associe à l'équation différentielle (1.7), l'équation dite homogène où sans second membre

$$a(t)y' + b(t)y = 0$$

qui est une équation différentielle à variables séparables.

1.6.2 Équations à variables séparables

Ce sont des équations du premier ordre sous forme normale données par l'équation

$$y' = f(t, y)$$

On peut regrouper (t, dt) d'une part et (y, dy) d'autre part. Le but est d'exprimer $f(t, y)$ sous la forme $g(t).h(y)$ Ce qui permettra de résoudre une équation du type

$$y' = g(t).h(y) \quad (1.8)$$

1.6.3 Équations de Bernoulli

Une équation différentielle de Bernoulli est une équation de la forme :

$$y' + a(t)y = b(t)y^m \quad (1.9)$$

Où $a(t)$, $b(t)$ sont des fonctions en t données et m une constante réelle différente de 0 et 1.

Méthode de résolution :

En divisant les deux membres de l'équation par y^m , on obtient :

$$\frac{y'}{y^m} + \frac{a(t)}{y^{m-1}} = b(t) \quad (1.10)$$

En effectuant le changement de fonction

$$z = \frac{1}{y^{m-1}}$$

donc

$$z' = (1-m) \frac{y'}{y^m}$$

L'équation (1.10) devient

$$\frac{z'}{(1-m)} + a(t)z = b(t)$$

Donc l'équation (1.10) se ramène à une équation différentielle linéaire du 1^{ère} ordre en z .

1.6.4 Équations de Riccati :

Une équation différentielle de Riccati est une équation de la forme :

$$y' = a(t)y^2 + b(t)y + c(t) \quad (1.11)$$

Où $a(t)$, $b(t)$ et $c(t)$ sont des fonctions en t données.

Méthode de résolution

On ne peut résoudre ce type d'équation que si l'on connaît une solution particulière y_1 . Avec le changement de fonction $y = y_1 + z$ et sachant que y_1 est solution particulière, on a :

$$y_1' = a(t)y_1^2 + b(t)y_1 + c(t)$$

L'équation (1.11) devient

$$z' = a(t)z^2 + [2a(t)y_1 + b(t)]z \quad (1.12)$$

1.7 Méthodes numérique de résolution des équations différentielles

Les méthodes numériques de résolution d'équations différentielles ordinaires sont nombreuses. On se contente dans cette section de présenter les méthodes à un pas parmi elles la méthode d'Euler et la méthode de Runge-Kutta.

1.7.1 Formulation générale :

On commence en remarquant qu'une solution formelle de l'équation :

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

est obtenue en intégrant de t_0 à t :

$$\int_{y(t_0)}^{y(t)} dy = \int_{t_0}^T f(s, y(s)) ds$$

$$y(t) - y(t_0) = \int_{t_0}^T f(s, y(s)) ds$$

donc la solution $y(t)$ s'écrit

$$y(t) = y_0 + \int_{t_0}^T f(s, y(s)) ds$$

où $y_0 \equiv y(t_0)$ Le problème avec cette solution formelle est que l'inconnue $y(t)$ se trouve sous une intégrale.

1.7.2 Méthode d'Euler

La méthode d'Euler est la méthode la plus simple pour résoudre numériquement une équation différentielle ordinaire. Elle possède une belle interprétation géométrique et son emploi est facile. Toutefois, elle est relativement peu utilisée en raison de sa faible précision. Considérons le problème :

$$y' = f(t, y(t)), y(0) = y_0, t \in [0, T] \quad (1.13)$$

Les méthodes basées sur la série de Taylor se prêtent bien à des traitements numériques. Commençons par subdiviser l'intervalle $[t_0, t_0 + T]$ en N sous intervalles de même longueur h et appelons t_n les points de subdivision. Nous avons donc :

$$0 = t_0 < t_1 < \dots < t_n < t_{n+1} < \dots < t_N = T \quad (1.14)$$

et posons

$$h_n = t_{n+1} - t_n, 0 \leq n \leq N - 1 \quad (1.15)$$

La solution de (1.13) vérifié pour $0 \leq n \leq N - 1$:

$$y(t_{n+1}) = y(t_n) + \int_{t_n}^{t_{n+1}} f(s, y(s)) ds \quad (1.16)$$

On construit par récurrence une approximation y_n et $y(t_n)$ en remplaçant la relation précédente par :

$$y_{n+1} = y_n + h f(t_n, y_n) \quad n = 0, 1, \dots, N - 1 \quad (1.17)$$

Ce qui revient à approcher, pour $s \in]t_n, t_{n+1}[$, $f(s, y(s))$ par $f(t_n, y_n)$

1.7.3 Méthode de Runge-Kutta d'ordre 2

Les méthodes de Runge-Kutta sont des méthodes d'analyse numérique d'approximation de solutions des équations différentielles, elles ont été nommées ainsi en l'honneur des mathématiciens Carl Runge (1856-1927) et Martin Wilhelm Kutta (1867-1944) lesquels élaborèrent la méthode en 1901. Ces méthodes reposent sur le principe de l'itération, c'est-à-dire qu'une première estimation de la solution est utilisée pour calculer une seconde estimation plus précise et ainsi de suite.

On considère le problème de Cauchy :

$$\begin{cases} y'(t) = f(t, y(t)) \\ y(0) = y_0 \end{cases}$$

On cherche à discrétiser ce problème par rapport à une subdivision :

$$0 = t_0 < t_1 < \dots < t_n < t_{n+1} < \dots < t_N = T$$

L'idée est de calculer par récurrence les points : $(t_n, y(t_n))$

posons :

$$h_n = t_{n+1} - t_n, \quad h = \max h_n, \quad 0 \leq n \leq N - 1$$

La méthode de Runge-Kutta correspondante est définie par l'algorithme suivant :

$$\begin{cases} k_1 = h \times f(t_{n-1}, y_{n-1}) \\ k_2 = h \times f(t_{n-1} + \frac{h}{2}, y_{n-1} + \frac{k_1}{2}) \\ y(t_n) = y(t_{n-1}) + k_2 \end{cases}$$

1.7.4 Méthode de Runge-Kutta d'ordre 4

La technique Runge-Kutta à l'ordre 4 intervient dans la plupart des programmes EDO (Équations différentielles ordinaires) comme ceux utilisés par Matlab et Octave. La formule Runge-Kutta à l'ordre 4 la plus utilisée, elle a une forme assez symétrique définie comme suit :

$$\begin{cases} k_1 = h \times f(t_{n-1}, y(t_{n-1})) \\ k_2 = h \times f(t_{n-1} + \frac{1}{2}h, y(t_{n-1}) + \frac{1}{2}k_1) \\ k_3 = h \times f(t_{n-1} + \frac{1}{2}h, y(t_{n-1}) + \frac{1}{2}k_2) \\ k_4 = h \times f(t_{n-1} + h, y(t_{n-1}) + \frac{1}{2}k_3) \\ y(t_n) = y(t_{n-1}) + (k_1 + 2k_2 + 2k_3 + k_4) \frac{1}{6} \end{cases}$$

L'interprétation géométrique des variantes k_1, k_2, k_3, k_4 :

k_1 : La pente au début de l'intervalle.
 k_2 : La pente au milieu de l'intervalle, en utilisant la pente k_1 pour calculer la valeur de y au point $t_n + \frac{h}{2}$ par le biais de la méthode d'Euler.
 k_3 : de nouveau, la pente au milieu de l'intervalle, mais obtenue cette fois en utilisant la pente k_2 pour calculer y .
 k_4 : la pente à la fin de l'intervalle, avec la valeur y calculée en utilisant k_3 .
 y_{n+1} : le schéma de la méthode RK4, ou-bien la moyenne des quatre pentes en affectant des poids plus grands à ceux des points milieu.

1.8 Conclusion

La plupart des grands mathématicien ont touché de près où de loin aux équations différentielles et le nombre où elles interviennent ne serait qu'un vaste catalogue de 80% des mathématiques et de la physique moderne, Grâce au développement des méthodes numériques on arrive souvent à mesurer où à estimer les résultats de petites variations d'un où plusieurs paramètres pendant un laps de temps suffisamment court pour que l'on puisse interpréter les variations de ces paramètres comme leurs dérivées.

Chapitre 2

Théorie du Contrôle Optimale

2.1 Introduction

En mathématique, la théorie du contrôle optimal s'inscrit dans la continuité du calcul des variations. Cette méthode est apparue après la seconde guerre mondiale, répondant à des besoins pratiques de guidage, notamment dans le domaine de l'aéronautique et de la dynamique. La formalisation de cette théorie a posée des questions nouvelles ; par exemple dans la théorie des équations différentielles ordinaires, elle a motivée un concept de solutions généralisées et a engendré de nouveaux résultats d'existence de trajectoires optimales. La théorie du contrôle optimal est très liée à la mécanique classique, en particulier aux principes de Fermat, de Huygens, équations d'Euler-Lagrange

2.2 Définitions :

- Les variables nommées variables d'état seront notées x_i , $i = 1, \dots, n$
- Les systèmes considérés évoluent dans le temps, donc les x_i sont des fonctions de temps notées : $x_i(t)$; $t \in [0, T]$.
- Les variables de contrôle, appelées aussi variables d'entrée, seront notées $u_j(t)$, $j = 1, \dots, m$.
- Elles doivent être intégrables par rapport à t , chose qui est bien souvent trop restrictive car ces fonctions peuvent être continues par morceaux ou de type Bang-Bang.
- Les n variables d'état vont être gouvernées par n équations différentielles du premier ordre nommées équations d'état de la forme :

$$\dot{x}(t) = f(t, x, u), \quad t \in [0, T] \quad (2.1)$$

ou \dot{x} est le vecteur dérivé par rapport au temps t de toutes les composantes de x .

- L'ensemble U est l'ensemble des contrôles admissibles qui peut être non bornée, bornée ou du type Bang-Bang.

Commande bornée :

La commande $u_j(t)$ est dite commande bornée si elle peut être minorée et majorée par des constantes a_j et b_j , par la forme suivante :

$$a_j \leq u_j(t) \leq b_j, \quad j = 1..m, \quad t \in [0, T]$$

Commande de Bang Bang :

Un contrôle $u \in U$ est appelé un contrôle de Bang Bang si pour chaque instant t et chaque et chaque indice $j = 1, \dots, m$ on a :

$$|u_j(t)| = 1$$

Commande admissible :

Le contrôle $u(t)$, $t \in [0, T]$ est dit admissible s'il satisfait les contraintes :

$$\begin{cases} \dot{x} = f(t, x(t), u(t)), & x(0) = x_0 \\ d_* \leq u(t) \leq d^* \end{cases} \quad (2, 2)$$

Commande continue par morceaux :

Une commande $u(t)$, $t_0 \leq t \leq T$ est dite continue par morceaux, notée $u \in \hat{C}[t_0, T]$, s'il existe une partition $t_0 < t_1 < \dots < t_N < t_{N+1} = T$ tel que u peut être considéré comme une fonction continue dans $[t_k, t_{k+1}]$ pour chaque $k = 0, 1, \dots, N$.

2.3 Positions du problème

La formulation du problème de contrôle optimal est le suivant :

$$J(T, u) = g(T, x(T)) + \int_0^T (f^0(t, x(t), u(t))) dt \rightarrow \min_u, \quad (2.3)$$

$$\dot{x} = f(t, x(t), u(t)), \quad (2.4)$$

$$x(0) = x_0 \in M_0, \quad (2.5)$$

$$x(T) = x_1 \in M_1, \quad (2.6)$$

$$u \in U \quad t \in I = [0, T], \quad (2.7)$$

Ou M_0 (ensemble de départ) et M_1 (ensemble d'arrivée) sont des sous-ensembles de \mathbb{R}^n , I intervalle de \mathbb{R} , $x(0) = x_0$ est la position initial du système (2,3)-(2.7) $x(T)$ est sa position terminale. En pratique, la position du système peut représenter la vitesse, la position, la température,... etc.

$u(t)$ est la commande du système (2.3)-(2.7). U est l'ensemble des applications (contrôles) mesurables, localement bornées sur I à valeurs dans \mathbb{R}^n

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

est appelé coût, critère de qualité ou but du problème (2.3)-(2.7). Ce critère de performance, est généralement décrit sous la forme de Bolza (Mayer Lagrange). On distingue deux variantes :

2.3.1 Problème de Lagrange :

C'est le problème dont le critère à minimiser s'écrit :

$$J(T, u) = \int_0^T (f^0(t, x(t), u(t))) dt \rightarrow \min_u, \quad (2, 8)$$

C'est à dire ($g \equiv 0$)

2.3.2 Problème de Mayer :

Ici c'est le problème dont le critère s'écrit :

$$J(T, u) = g(T, x(T)) \rightarrow \min_u, \quad (2, 9)$$

C'est à dire $f^0(t, x(t), u(t)) = 0$.

2.3.3 Classement des problèmes de contrôle optimal

On peut classer les fonctions objectif en deux critères physiques de performance :

Temps optimal :

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

Coût optimal :

On parle d'un problème en coût optimal lorsque le temps final T est fixé dans l'expression $\min_u J(T, u) = g(T, x(T)) + \int_0^T f^0(t, x(t), u(t)) dt$.

Avant de résoudre un problème de contrôle optimal, on se pose les questions suivantes :

Question 1 :

Existe-t-il un contrôle $u \in U$ tel que la trajectoire associée à $x(t)$ joigne l'état initial $x_0 \in \mathbb{R}^n$ à un état final $x_1 \in \mathbb{R}^n$ en un temps fini ?? C'est le problème de contrôlabilité.

Question 2 :

Si le système est contrôlable c'est à dire une fois la contrôlabilité est vérifiée, on peut vouloir déterminer un contrôle $u \in U$ tel que la trajectoire associée à ce contrôle joigne l'état initial $x_0 \in \mathbb{R}^n$ à un état terminal $x_1 = x(T) \in \mathbb{R}^n$, qui minimise un certain critère de performance.

2.4 Contrôlabilité :

2.4.1 Notion de contrôlabilité

La contrôlabilité est l'un des concepts centraux de la théorie du contrôle. C'est la possibilité d'influencer l'état du système (sortie) en manipulant les entrées (commandes). Pour déterminer une trajectoire optimale joignant un ensemble initial M_0 à une cible M_1 , il faut d'abord vérifier si cette cible est atteignable : c'est le problème de contrôlabilité. Existe-t-il un contrôle u tel que la trajectoire associée $x(t)$ conduit le système de M_0 à M_1 en un temps fini ?

La notion de contrôlabilité a été inventée en 1960 par Kalman [4] à propos des systèmes linéaires de la forme $\dot{x} = Ax + Bu$. L'état x évolue dans un espace vectoriel réel E , de dimension n .

On dit que $\dot{x} = Ax + Bu$ est contrôlable, si l'on peut joindre deux points de l'espace d'état, c'est à dire si, et seulement si, étant donnés deux points $x_0, x_1 \in E$ et deux instants t_0, t_1 avec $t_0 < t_1$, il existe une commande u , définie sur $[t_0, t_1]$ tel que : $x(t_0) = x_0, x(t_1) = x_1$.

2.4.2 Ensemble d'accessibilité

Considérons le système suivant

$$\dot{x} = f(t, x(t), u(t)) \quad x(0) = x_0 \quad t \in I = [0, T], \quad (\star)$$

-L'ensemble des points accessibles à partir de x_0 en un temps T est définie par :

$$Acc(x_0, T) = \{x_u(T) \in \mathbb{R}^n \mid u \in U\}$$

ou $x_u(T)$ est la solution du système associée au contrôle u , autrement dit $Acc(x_0, T)$ est l'ensemble des extrémités des solutions du système (\star) , en temps T lorsque le contrôle u varie.

-Le système de contrôle (\star) est dit contrôlable en un temps T si $Acc(x_0, T) = \mathbb{R}^n$ c à d : $\forall x_0, x_1 \in \mathbb{R}^n \exists u \in U$ telle que la trajectoire $x(t)$ associée à $u(t)$ relie x_0 à x_1 en un temps T fini.



FIG 1.1-Ensemble d'accisibilité

-Le contrôle u est dit extrémale sur $[0, T]$, si la trajectoire du système $\dot{x} = f(t, x(t), u(t))$ associée à u vérifie :

$$x(t) \in \partial Acc(x_0, T), \quad t \in I = [0, T], \quad (2.10).$$

2.4.3 Contrôlabilité des systèmes linéaires :

La formulation mathématique d'un système de contrôle linéaire est la suivante :

$$\dot{x} = A(t)x(t) + B(t)u(t) + r(t), \quad x(0) = x_0, \quad t \in I = [0, T], \quad (2.11)$$

Où I est un intervalle de \mathbb{R} . A , B et r sont trois applications localement intégrales sur I à valeurs respectivement dans $M_n(\mathbb{R})$, $M_{n,m}(\mathbb{R})$ et \mathbb{R} . l'ensemble des contrôles u considérés est l'ensemble des applications mesurables localement bornées sur I à valeurs dans un sous ensemble $U \subset \mathbb{R}$.

Soit $M(.) : I \longrightarrow M_n(\mathbb{R})$, la résolvante du système linéaire homogène $\dot{x} = A(t)x(t)$, définie par :

$$\begin{cases} \dot{M}(t) = A(t).M(t) \\ M(0) = id \end{cases}$$

ou id est la matrice identité.

Pour tout contrôle u le système $\dot{x} = A(t)x(t) + B(t)u(t) + r(t)$; $x(0) = x_0$ admet une unique solution $x(.) : I \longrightarrow \mathbb{R}^n$ absolument continue donnée par :

$$x(t) = M(t)x_0 + \int_0^T [M(t)M(s)^{-1}B(s)u(s) + r(s)]ds \quad t \in [0, T], \quad (2.12).$$

Si $r = 0$ et $x_0 = 0$, alors la solution du système s'écrit :

$$x(t) = M(t) \int_0^T M(s)^{-1}B(s)u(s)ds, \quad (2.13).$$

Elle est linéaire en u .

le théorème suivant donne une condition générale pour la contrôlabilité des systèmes linéaires :

Théorème 2.4.1 [2]

Le système $\dot{x} = A(t)x(t) + B(t)u(t) + r(t)$ est contrôlable en temps T si et seulement si la matrice :

$$C(t_f) = \int_0^T M(t)^{-1} B(t) B'(t) M(t)^{-1} dt, \quad (2.14).$$

dite matrice de contrôlabilité, est inversible.

$B'(t)$ est la transposée de $B(t)$ et $M^{-1}(t)$ est l'inverse de $M(t)$, C est appelée matrice de contrôlabilité.

2.4.4 Contrôlabilité des systèmes linéaires autonome :

Le système $\dot{x} = A(t)x(t) + B(t)u(t) + r(t)$ est dit autonome lorsque les matrices A et B ne dépendent pas de t . Dans ce cas, la matrice $M(t) = e^{tA}$.

La solution du système associée au contrôle u s'écrit, pour tout $t \in I$:

$$x(t) = e^{tA} \left(x_0 + \int_0^T e^{-As} (B(u) + r(s)) ds \right), \quad (2.15).$$

Le théorème suivant donne une condition nécessaire et suffisante de contrôlabilité dans le cas sans contrainte sur le contrôle.

Théorème 2.4.2 [2]

On suppose que $U = \mathbb{R}^m$. Le système $\dot{x} = A(t)x(t) + B(t)u(t) + r(t)$ est contrôlable en temps T si et seulement si la matrice :

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

et de rang n .

Où C est une matrice à n lignes et $n \times m$ colonnes, elle est appelée matrice de Kalman. La condition $\text{rang } C = n$ est appelée condition de Kalman.

2.4.5 Contrôlabilité des systèmes non-linéaires

La contrôlabilité est un concept clé pour la compréhension des propriétés structurelles et qualitatives, comme la stabilisation.

L'extension de la contrôlabilité au cas non-linéaire de dimension finie et de dimension infinie a suscité depuis près de cinquante ans une littérature considérable, qui n'a en rien épuisé ce sujet riche et varié.

Les auteurs, dans leur quasi-totalité, ont considéré des généralisations naturelles de :

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

Le résultat suivant donne une condition sur la contrôlabilité locale des systèmes non-linéaires :

proposition 2.5.1[2]

Considérons le système $\dot{x} = f((t, x(t), u(t)); x(0) = x^0$, avec $f(x^0, u^0) = 0$. On note $A = \frac{\partial f}{\partial x}(x^0, u^0)$ et $B = \frac{\partial f}{\partial u}(x^0, u^0)$.

Si $\text{rang } C = [B, |AB|, \dots, |A^{n-1}B|] = n$, alors le système est localement contrôlable en x^0 . En général, le problème de contrôlabilité globale est difficile. Cependant, il existe des techniques qui permettent de déduire la contrôlabilité locale dans le cas des systèmes linéaires.

2.5 Principe du maximum :

Dans cette section on donne une version générale du principe du maximum de Pontryagin. Ce théorème est difficile à démontrer. En revanche lorsqu'il n'y a pas de contrainte sur le contrôle, la preuve est simple, et on arrive au principe du maximum dit faible. C'est à cette version plus simple que nous allons d'abord nous intéresser. Puis nous passerons au cas général.

2.5.1 Principe du maximum faible :

a - Le problème de Lagrange :

Dans ce problème on cherche des conditions nécessaires d'optimalité pour le système :

$$\dot{x}(t) = f(t, x(t), u(t)), \quad (2, 16)$$

Où les contrôles $u(.) \in U$ sont définis sur $[t_0, T]$ et les trajectoires associées doivent vérifier $x(t_0) = x_0$ et $x(t_f) = x_1$, le problème est d'optimiser un coût de la forme :

$$J(u) = \int_{t_0}^{t_f} f^0(t, x(t), u(t)) dt, \quad (2, 17)$$

Où T est fixé.

Associons au système (2.17) le système augmenté suivant :

$$\begin{cases} \dot{x}(t) = f(t, x(t), u(t)) \\ \dot{x}^0(t) = f^0(t, x(t), u(t)) \end{cases}, \quad (2, 18)$$

et notons $\tilde{x} = (x, x^0)$, $\tilde{f} = (f, f^0)$. Le problème revient donc à chercher une trajectoire solution de (2.16) joignant les points $\tilde{x}_0 = (x, x^0)$ et $\tilde{x}_1 = (x_1, x^0(t_f))$, et minimiser la dernière coordonnée $x^0(T)$.

Théorème 2.5.1[2]

Si le contrôle u associé au système de contrôle (2.16) est optimal pour le coût (2.17), alors il existe une application $p(\cdot)$ absolument continue sur $[t_0, T]$, à valeur dans \mathbb{R}^n , appelée vecteur adjoint, et un réel $p^0 \leq 0$ tels que le couple $(p(\cdot), p^0)$ est non trivial, les équations suivantes sont vérifiées pour presque tout $t \in [t_0, T]$.

$$\dot{x}(t) = \frac{\partial H}{\partial p}(t, x(t), p(t), p^0, u(t)), \quad (2, 19)$$

$$\dot{p}(t) = -\frac{\partial H}{\partial x}(t, x(t), p(t), p^0, u(t)), \quad (2, 20)$$

$$\frac{\partial H}{\partial u}(t, x(t), p(t), p^0, u(t)) = 0, \quad (2.21)$$

Où H est le Hamiltonien associé au système (2.16) et au (2.17)

$$H(t, x, p, p^0, u) = \prec(p, f(t, x, u)) \succ + p^0 f(t, x, u), \quad (2.22)$$

b-Problème de Mayer-Lagrange :

On modifie le problème précédent on introduisant le coût

$$J(x, t) = g(t, x_u(t)) + \int_{t_0}^T f^0(s, x_u(s), u(s)) ds, \quad (2, 23)$$

et où le temps final t n'est pas fixé. Soit M_1 une variété de \mathbb{R}^n . Le problème de contrôle optimal est alors de déterminer une trajectoire solution de :

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

Où les contrôles $u(\cdot)$ sont dans l'ensemble U des contrôles admissibles sur $[t_0, T]$ tel que $x(T) \in M_1$, et de plus $x(\cdot)$ minimise sur $[t_0, T]$ le coût (2.23).

Théorème 2.5.1 [3]

Si le contrôle u est optimal sur $[t_0, T]$ alors il existe une application $p : [t_0, T] \rightarrow \mathbb{R}^n$ absolument continue, et un réel $p^0 \leq 0$, tels que le couple $(p(\cdot), p^0)$ est non trivial, et

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

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

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

Où H est le Hamiltonien au système (2.16) et au coût (2.23)

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

2.5.2 Principe de maximum de Pontryagin (PMP) :

Le principe du maximum de Pontryagin énonce que la trajectoire optimale vérifie les trois conditions suivantes :

a- L'équation adjoint :

$$p(t_f) = p_0 \frac{\partial g(x(t_f))}{\partial x(t_f)}, \quad (2, 24)$$

Dans le cas où l'état final est libre, une condition apparaît sur la valeur finale de l'état adjoint :

$$p(t_f) = p_0 \frac{\partial g(x(t_f))}{\partial x(t_f)}, \quad (2.25)$$

b- Une condition qui suppose qu'aucune contrainte n'est subie par la commande, c'est la condition de stationnarité :

$$\frac{\partial H(x, u, p, t)}{\partial u} = 0, \quad (2.26)$$

c- L'équation d'état

$$\dot{x} = \frac{\partial H(x, u, p, t)}{\partial p} \text{ avec } x(t_0) = x_0$$

Remarque 2.5.1 :

La convention $p^0 < 0$ conduit au principe du maximum. La convention $p^0 > 0$ conduirait au principe du minimum.

2.5.3 Condition de transversalité :

La condition de transversalité est une contrainte supplémentaire (ou une condition finale), qui est énoncée sur l'état adjoint dans le cas où l'état final est libre. La formule de cette condition est donnée par les équations (2.24)(2.25).

2.5.4 Exemple :

Soit le système continu du premier ordre défini par l'équation différentielle :

$$\dot{x} = -x + u$$

que l'on désire amener au bout d'un temps t_f donné, d'un état initial x_0 donné à un état final voisin de zéro, en minimisant, l'énergie de la commande. Pour réaliser cet objectif, on considère le critère à minimiser :

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

1. Hamiltonien :

$$H = p(-x + u) - \frac{1}{2}u^2$$

2. Conditions d'optimalité :

$$\dot{p} = -\frac{\partial H}{\partial u} = p \implies p - \dot{p} = 0$$

$$\frac{\partial H}{\partial u} = p - x = 0 \implies u^0 = p$$

Remarque 2.5.2 :

$\frac{\partial^2 H}{\partial u^2} = -1$, $u^0 = p$ c'est bien un maximum.

Transversalité initiale : x_0 donné, p_0 inconnu .

Transversalité finale : inconnu,

critère terminal $\frac{1}{2}x^2(T)$ implique $p(T) = -x(T)$.

3. Intégration

a) Du système adjoint :

$$p = p_0 e^t$$

b) Du processus : en remplaçant u par la commande optimale, il s'écrit :

$$\dot{x} + x = p_0 e^t$$

D'où

$$x = (x_0 - \frac{1}{2}p_0)e^{-t} + \frac{1}{2}p_0 e^t$$

$$\dot{x} = -(x_0 - \frac{1}{2}p_0)e^{-t} + \frac{1}{2}p_0 e^t$$

4. La condition de transversalité finale $p(T) + x(T) = 0$ permet d'écrire :

$$p_0 e^T + (x_0 - \frac{1}{2}p_0)e^{-T} + \frac{1}{2}p_0 e^T = 0$$

D'où :

$$p_0 = \frac{2}{1 - 3e^{2T}} x_0$$

5. La commande optimale :

$$u^0(t) = \frac{2}{1 - 3e^{2(T-t)}}x(t)$$

2.6 Méthodes numériques en contrôle optimal :

On distingue deux types de méthodes numériques en contrôle optimal [2] : les méthodes directes et les méthodes indirectes. Les méthodes directes consistent à discrétiser l'état et le contrôle, et réduisent le problème à un problème d'optimisation non linéaire programmation non linéaire, ou "nonlinear programming". Les méthodes indirectes consistent à résoudre numériquement, par une méthode de tir "shooting method", un problème aux valeurs limites obtenu par application du principe du maximum.

2.6.1 Méthodes directes

La discrétisation totale

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 (ou problème de programmation non linéaire) de la forme

$$\min_{Z \in C} F(Z), \quad (2, 27)$$

Où $Z(x, \dots, x_N, u_1, \dots, u_N)$, et

$$C = \{Z \mid g_i(Z) = 0, i \in 1 \dots r, g_j(Z) \leq 0, j \in r + 1 \dots m\}, \quad (2, 28)$$

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 $0 = t_0 < t_1 < \dots < t_N = T$ de l'intervalle $[0, T]$. Réduisons l'espace des contrôles en considérant (par exemple) des contrôles constants par morceaux selon cette subdivision. Par ailleurs, choisissons une discrétisation de l'équation différentielle, par exemple choisissons ici (pour simplifier) la méthode d'Euler explicite. 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 2.6.1 :

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. D'autre part, il existe de nombreuses méthodes pour discrétiser une équation différentielle ordinaire : méthode d'Euler (explicite ou implicite), point milieu, Runge-Kutta. Le choix de la méthode dépend du problème abordé. La discrétisation précédente conduit donc au problème de programmation non linéaire

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

$$\min C(x, \dots, x_N, u_1, \dots, u_N),$$

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

i.e. un problème du type (2.27).

Remarque 2.6.2 :

Cette méthode est très simple à mettre en œuvre. 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 :

$$u \in Vect(U_1, \dots, U_N), \text{ i.e } u(t) = \sum_{i=1}^N u_i U_i(t), \quad u_i \in \mathbb{R}$$

$$x \in Vect(X_1, \dots, X_N), \text{ i.e } x(t) = \sum_{i=1}^N x_i X_i, \quad x_i \in \mathbb{R}$$

Où les $U_i(t)$ et $X_i(t)$ représentent une base de Galerkin. 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 (2.27).

La résolution numérique d'un problème de programmation non linéaire du type (2.27) est standard. Elle peut être effectuée, par exemple, par une méthode de pénalisation, ou par une méthode SQP (sequential quadratic programming).

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 (2.27), (2.28), les conditions de Kuhn-Tucker s'écrivent

$$\nabla F(Z) + \sum_{i=1}^m \lambda_i \nabla g_i(Z) = 0$$

Où les multiplicateurs de Lagrange λ_i vérifient

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

Les méthodes SQP consistent à calculer de manière itérative ces multiplicateurs de Lagrange, en utilisant des méthodes de Newton ou quasi-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.

2.6.2 Méthodes indirectes

La méthode de tir simple

Considérons le problème de contrôle optimal (2.2), (2.3), et supposons dans un premier temps que le temps final T 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 :

$$\dot{z}(t) = F(t, z(t)), \text{ où } z(t) = (x(t), p(t))$$

et les conditions initiales, finales, et les conditions de transversalité, se mettent sous la forme $R(z(0), z(t_f)) = 0$. Finalement, on obtient le problème aux valeurs limites :

$$\begin{cases} \dot{z}(t) = F(t, z(t)) \\ R(z(0), z(t_f)) = 0, \end{cases} \quad (2.29)$$

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

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

et posons $G(z_0) = R(z_0, z(t_f, z_0))$ le problème (2.29) aux valeurs limites est alors équivalent

$$G(z_0) = 0$$

i.e. il s'agit de déterminer un zéro de la fonction G . Ceci peut se résoudre par une méthode de Newton.

Remarque 2.6.1 :

Si le temps final T est libre, on peut se ramener à la formulation précédente en considérant T comme une inconnue auxiliaire. On augmente alors la dimension de l'état en considérant l'équation supplémentaire $\frac{dT}{dt} = 0$. On peut utiliser le même artifice si le contrôle est bang-bang, pour déterminer les temps de commutation. Il peut cependant s'avérer préférable, lorsque le temps final est libre, d'utiliser la condition de transversalité sur le Hamiltonien.

Méthode de tir multiple

Par rapport à la méthode de tir simple, la méthode de tir multiple découpe l'intervalle $[0, T]$ 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. De manière plus précise, considérons un problème de contrôle optimal général. L'application du principe du maximum réduit le problème à un problème aux valeurs limites du type

$$\dot{z}(t) = F(t, z(t)) = \begin{cases} F_0(t, z(t)) & \text{si } t_0 \leq t < t_1 \\ F_1(t, z(t)) & \text{si } t_1 \leq t < t_2, \\ \vdots \\ F_s(t, z(t)) & \text{si } t_s \leq t \leq T \end{cases} \quad (2.30)$$

Où $z = (x, p) \in R^{2n}$ (p est le vecteur adjoint), et $t_1, t_2, \dots, t_s \in [t_0, T]$ 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.

Remarque 2.6.2 :

Apriori le temps final T 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 $[t_0, T]$ en N sous intervalles, la valeur de $z(t)$ au début de chaque sous-intervalle étant inconnue. Plus précisément, soit $t_0 < \sigma_1 < \dots < \sigma_k < T$ une subdivision fixée de l'intervalle $[t_0, T]$. En tout point σ_i la fonction z est continue. On peut considérer σ_i comme un point de commutation fixe, en lequel on a

$$\begin{cases} z(\sigma_j^+) = z(\sigma_j^-); \\ \sigma_j = \sigma_j^* \text{ fixé.} \end{cases}$$

On définit maintenant les nœuds

$$\{\tau_1, \dots, \tau_m\} = \{t_0, T\} \cup \{\sigma_1, \dots, \sigma_k\} \cup \{t_1, \dots, t_s\}$$

Finalement on est conduit au problème aux valeurs limites

$$\begin{aligned} \bullet \dot{z}(t) = F(t, z(t)) &= \begin{cases} F_0(t, z(t)) & \text{si } \tau_1 \leq t < \tau_2 \\ F_1(t, z(t)) & \text{si } \tau_2 \leq t < \tau_3 \\ \vdots \\ F_{m-1}(t, z(t)) & \text{si } \tau_{m-1} \leq t \leq \tau_m \end{cases} \\ \bullet \forall j \in \{2, \dots, m-1\} \quad r_j(\tau_j, z(\tau_j^-), z(\tau_j^+)) &= 0 \end{aligned}$$

$$\bullet r_m(\tau_m, z(\tau_1), z(\tau_m)) = 0$$

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

Remarque 2.6.3 :

On améliore la stabilité de la méthode en augmentant le nombre de nœuds. 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 $T - t_0$. 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 (2,30) peut se paralléliser.

Posons $z_j^+ = z(\tau_j^+)$, et soit $z(t, \tau_{j-1}, z_{j-1}^+)$ la solution du problème de Cauchy

$$\dot{z}(t) = F(t, z(t), z(\tau_{j-1}) = z_{j-1}^+$$

on a

$$z(\tau_j^-) = z(\tau_j^-, \tau_{j-1}, z_{j-1}^+)$$

Les conditions intérieures et frontières s'écrivent

$$\forall j \in \{2, \dots, m-1\} \quad r_j(\tau_j, z(\tau_j^-, \tau_{j-1}, z_{j-1}^+), z_j^+) = 0, \quad (2, 31)$$

$$r_m = (\tau_m, z_1^+, z(\tau_j, z(\tau_m^-, \tau_{m-1}, z_{m-1}^+)) = 0$$

Posons maintenant

$$z = (z_1^+, \tau_m, z_2^+, \tau_2, \dots, z_{m-1}^+, \tau_{m-1})^T \in R^{(2n+1)(m-1)}$$

Où $z \in R^{2n}$. Alors les conditions (2,31) sont vérifiées si

$$G(z) = \begin{pmatrix} r_m = (\tau_m, z_1^+, z(\tau_j, z(\tau_m^-, \tau_{m-1}, z_{m-1}^+)) \\ r_2 = (\tau_2, z(\tau_2^-, \tau_1, z_1^+, z_2^+)) \\ \vdots \\ r_{m-1} = (\tau_m, z(\tau_{m-1}^-, \tau_{m-2}, z_{m-2}^+, z_{m-1}^+)) \end{pmatrix} = 0$$

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.

2.6.3 Quelle méthode choisir ?

Le tableau suivant résume les caractéristiques des méthodes directes et indirectes.

Méthodes directes	Méthodes indirectes
Mise en œuvre simple, sans connaissance à priori de la solution.	Connaissance à priori de la structure de la trajectoire optimale.
Facilité de la prise en compte des contraintes sur l'état.	Difficulté théorique de la prise en compte de contraintes sur l'état.
Peu sensible au choix de la condition initiale.	Très sensibles au choix de la condition initiale.
Contrôles (globalement) optimaux en boucle fermée.	Contrôles (localement) optimaux en boucle ouverte.
Précision numérique basse ou moyenne.	Très grande précision numérique.
Efficaces en basse dimension.	Efficaces en toute dimension.
gourmandise en mémoire.	calculs parallélisables.

2.7 Exemple :

Soit $\ddot{x} = u$ l'équation du mouvement d'un objet se déplaçant sur une trajectoire sans frottement, ou u est la force agissant sur l'objet vérifiant $|u| \leq 1$.

Les conditions initiales sont : $x(0) = 0$, $\dot{x}(0) = 0$, et les conditions finales sont $x(2) = 0.5$, $\dot{x}(2) = 0$.

On cherche la meilleure commande pour ramener l'objet du point initial $(0,0)$ au point final $(0.5,0)$ tout en minimisant la valeur $\int_0^2 u^2 dt$.

En procédant à un changement de variable :

$$x_1 = x$$

$$x_2 = \dot{x}$$

on obtient le problème de contrôle optimal formulé comme suit :

$$\left\{ \begin{array}{l} \min J(u) = \min \int_0^2 u^2 dt \\ sc \\ \dot{x}_1 = x_2 \\ \dot{x}_2 = u \\ |u| \leq 1 \\ x_1(0) = 0, \quad x_1(2) = 0.5 \\ x_2(0) = 0, \quad x_2(2) = 0 \end{array} \right.$$

2.7.1 Résolution par la méthode indirecte sous MatLab :

Le hamiltonien du système s'écrit :

$$H(x, z, u) = z_0 u^2 + z_1 x_2 + z_2 u$$

On maximise le hamiltonien H en utilisant le principe du maximum ($z_0 = -1$), on aura

$$\max_{-1 \leq v \leq 1} (-v^2 + z_2 v + z_1 x_2)$$

On obtient un contrôle optimal

$$u^* = \begin{cases} \frac{z_2}{2} & \text{si } \frac{|z_2|}{2} < 1 \\ \text{sign}(z_2) & \text{sinon} \end{cases}$$

Le système différentiel d'état et d'état adjoint s'écrit

$$\begin{cases} \dot{x}_1 = x_2, x_1(0) = 0, x_1(2) = 0.5 \\ \dot{x}_2 = u, x_2(0) = 0, x_2(2) = 0 \\ \dot{z}_1 = 0, z_1(0) \in \mathbb{R}, z_1(2) \in \mathbb{R} \\ \dot{z}_2 = -z_1, z_2(0) \in \mathbb{R}, z_2(2) \in \mathbb{R} \end{cases}$$

Le programme

```
1. function tir_simple1
2. clear all ; clf ; clc;
3. global x0 tf;
4. x0 = [0; 0] ; tf = 2; P0 = [0.5; 0.5];
5. %
6. options = optimset('display','iter','Algorithm','levenberg-marquardt');
7. [P0f, FVAL, EXITFLAG] = fsolve(@S, P0, options);
8. P0f, FVAL, EXITFLAG
9. Tracé des trajectoires optimales
10. options = odeset('AbsTol', 1e-9, 'RelTol', 1e-9);
11. [t, y] = ode45(@sys, [0; tf], [x0; P0f], options);
12. subplot(321); plot(t, y(:, 1)); title('Trajectoiredex1(t)'); grid on
13. subplot(322); plot(t, y(:, 2)); title('Trajectoiredex2(t)'); grid on
14. subplot(323); plot(t, y(:, 3)); title('Trajectoiredex1(t)'); grid on
15. subplot(324); plot(t, y(:, 4)); title('Trajectoiredex2(t)'); grid on
16. if abs(y(:, 4)) < 2
17. y(:, 5) = -y(:, 4)/2;
18. else
19. y(:, 5) = -sign(y(:, 4));
20. end
```

```

21. subplot(325); plot(t, y(:, 5)); title('Trajecoire de u(t)'); grid on
22. Définition de la fonction de tir
23. function Yzero = S(Y)
24. global x0 tf;
25. options = odeset('AbsTol', 1e-9, 'RelTol', 1e-9);
26. [t, y] = ode45(@sys, [0; tf], [x0; Y], options);
27. Yzero = [y(end, 1) - 0.5
28. y(end, 2)];
29. système extrêmeal
30. function ydot = sys(t, y)
31. if abs(y(4)) < 2
32. u = -y(4)/2;
33. else
34. u = -sign(y(4));
35. end
36. ydot = [y(2)
37. u
38. 0
39. -y(3)];

```

Les résultats sont donné dans les figures suivantes :

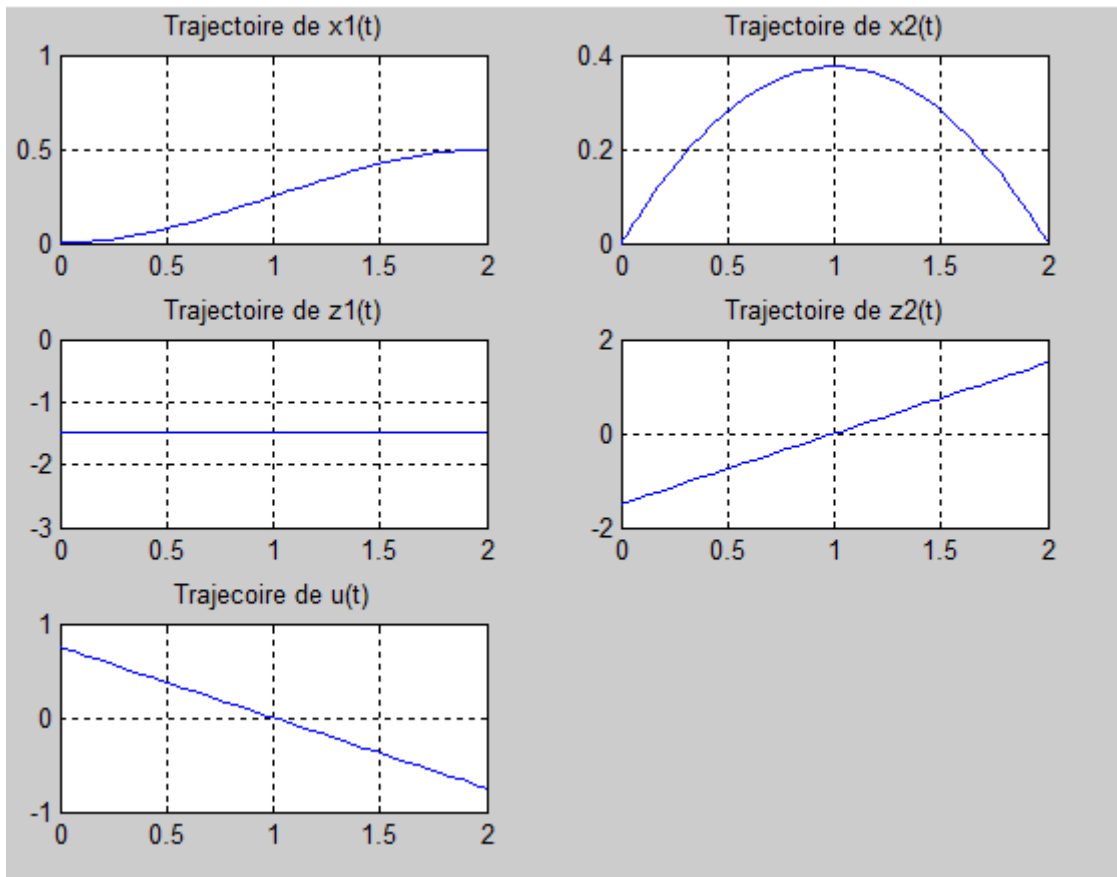


FIG 2.2 Résultats par la méthode indirecte

2.7.2 Résolution par la méthode directe sous MatLab :

On se ramène à un problème dont la résolution est effectuée sous Matlab, en utilisant l'intégrateur numérique RK4, implémenté avec la routine *fmincon.m*.

Le programme

```
1. function Direct_RK4
2. clear all; closeall; clc;
3. global tf
4. n = 100; tf = 2;
5. uinit = 2 * rand(n, 1) - 1;
6. xinit = uinit;
7. Lb = -ones(n, 1); ub = ones(n, 1);
8. options = optimset('Display','iter','Algorithm','active-set');
9. [res, Fval, exitflag] = fmincon(@fobj, rinit, [], [], [], [], Lb, ub, @cond, options);
10. fprintf('contrôle ');
11. res;
12. fprintf('EXITFLAG =
13. ===== tracédescourbesoptimales =====
14. x(1) = 0; y(1) = 0;
15. for i = 1 : n - 1
16. k11 = tf/n * y(i);
17. k21 = tf/n * (y(i) + k11/2);
18. k31 = tf/n * (y(i) + k21/2);
19. k41 = tf/n * (y(i) + k31/2);
20. x(i + 1) = x(i) + (k11 + 2 * k21 + 2 * k31 + k41)/6;
21. k12 = tf/n * res(i);
22. k22 = tf/n * (res(i) + k12/2);
23. k32 = tf/n * (res(i) + k22/2);
24. k42 = tf/n * (res(i) + k32);
25. y(i + 1) = y(i) + (k12 + 2 * k22 + 2 * k32 + k42)/6;
26. end
27. subplot(221); plot(linspace(0, tf, n), x(1 : n), 'b'); title('Trajectoiredex'); grid on
28. subplot(222); plot(linspace(0, tf, n), y(1 : n), 'b'); title('Trajectoiredey'); grid on
29. subplot(223); plot(linspace(0, tf, n), res(1 : n), 'b'); title('contrôle'); grid on
30. ===== contraintesduproblèmes =====
31. function[c, ceq] = cond(u)
32. global tf;
33. n = length(u);
34. xf = 0; yf = 0;
```



```

35.  $k_{11} = tf/n * yf$ ;
36.  $k_{21} = tf/n * (yf + k_{11}/2)$ ;
37.  $k_{31} = tf/n * (yf + k_{21})$ ;
38.  $k_{31} = tf/n * (yf + k_{21})$ ;
39.  $k_{31} = tf/n * (yf + k_{21})$ ;
40.  $xf = xf + (k_{11} + 2 * k_{21} + 2 * k_{31} + k_{41})/6$ ;
41.  $k_{12} = tf/n * u(i)$ ;
42.  $k_{22} = tf/n * (u(i) + k_{12}/2)$ ;
43.  $k_{32} = tf/n * (u(i) + k_{22}/2)$ ;
44.  $k_{42} = tf/n * (u(i) + k_{32})$ ;
45.  $yf = yf + (k_{12} + 2 * k_{22} + 2 * k_{32} + k_{42})/6$ ;
46. end
47.  $c = 0$ ;
48.  $ceq = [xf - 0.5, yf]$ ;
49. %fonction objectif
50. function val = fobj(u)
51. global tf;
52.  $n = length(u)$ ;
53.  $val = 0$ ;
54. for  $i = 1 : n$ 
55.  $k_{11} = (tf/n) * u(i)^2$ ;
56.  $k_{21} = (tf/n) * (u(i)^2 + k_{11}/2)$ ;
57.  $k_{31} = (tf/n) * (u(i)^2 + k_{21}/2)$ ;
58.  $k_{41} = (tf/n) * (u(i)^2 + k_{31})$ ;
59.  $val = val + (k_{11} + 2 * k_{21} + 2 * k_{31} + k_{41})/6$ ;
60. end

```

Les résultats sont donnés dans les figures suivantes :

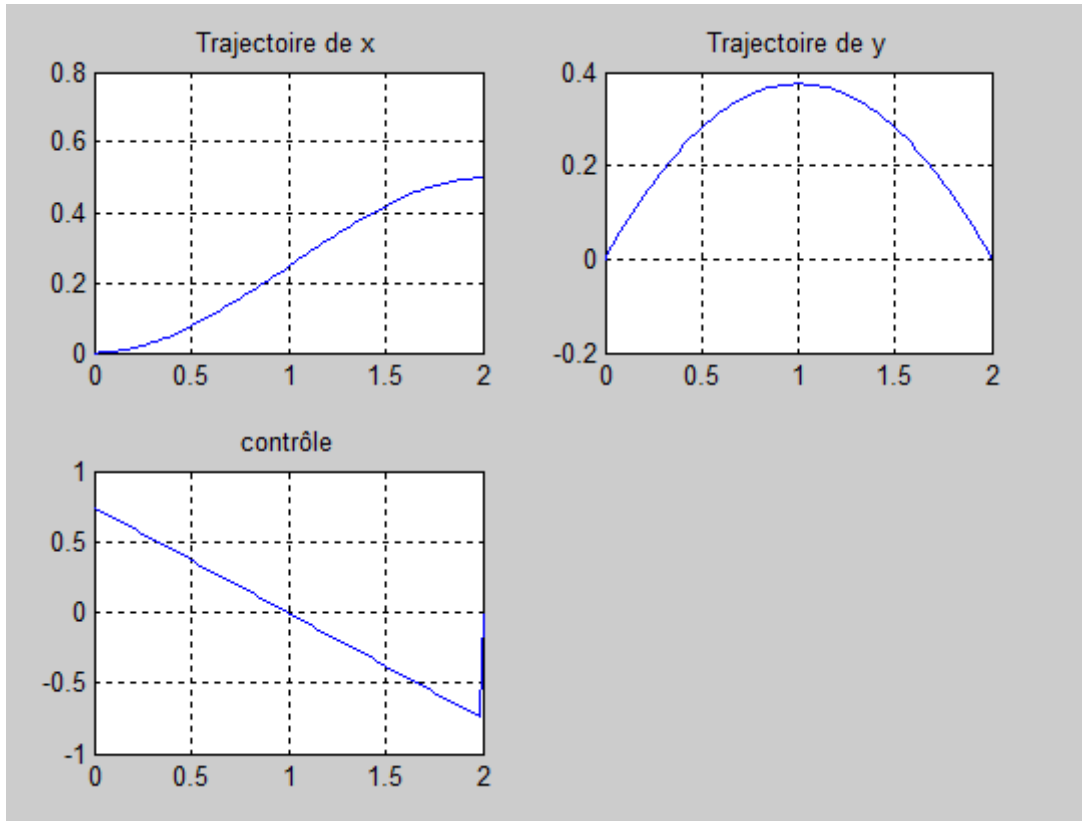


FIG 2.3 .Résultats par la méthode directe

2.8 Conclusion

Dans ce chapitre on a proposé une généralité sur la théorie de contrôle optimale qui a commencé dans les années 50, avec la formalisation du principe de maximum de Pontryagin et le développement des méthodes de résolution. 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 dans de nombreux domaines tels que l'aérospatiale (problèmes de transfert orbital de planification des vols), l'automatique et la robotique (problèmes de coordination de mouvements des robots), la biologie (contrôle des réseaux de régulation génique), ainsi que d'autres domaines.

Chapitre 3

Contrôle optimal du problème de GODDARD

3.1 Introduction

Au cours des années 1920, en Europe comme aux États-Unis, le désir de conquérir les espaces interplanétaires devient de plus en plus vif. Pour parvenir à cette fin, il faut disposer d'une fusée puissante utilisant des ergols liquides, plus énergétiques que les poudres, employées jusqu'alors. Cette problématique a inspiré plusieurs chercheurs. L'Américain Robert Hutchings Goddard (1882-1945) a été l'un des premiers à avoir créé et développé les fusées à carburant liquide. C'est à Auburn (Massachusetts, USA) un 16 mars 1926 qu'il se fut le premier lancement d'une fusée propulsée grâce à un mélange d'essence et d'oxygène liquide. En 2.5 secondes de vol seulement l'engin atteint une altitude de 12,50 mètres avec une vitesse de 100 km/h. Ce premier essai encouragea Goddard à façonner ses travaux et perfectionner le système de propulsion de ses fusées, la voie était dès lors ouverte à la réalisation d'engins plus puissants et plus performants.

le problème de Goddard a été formulé en 1919 [5], [6], [7], [8], [9], qui consiste à maximiser (minimiser) différents critères de performances (l'altitude, l'énergie, la vitesse, l'accélération, la masse), Ce problème devient aussitôt une référence dans la théorie du contrôle optimal en raison de la structure du modèle relativement simple, ce qui rend les fusées de Goddard un objet d'étude idéal.

Dans ce chapitre nous allons présenter le modèle de ce problème, nous procédons sa résolution par méthode indirecte puis la méthode directe suivie de leurs codes MATLAB.

3.2 Généralités :

3.2.1 La fusée :

Un véhicule capable d'échapper à l'attraction terrestre et de se déplacer dans l'espace, grâce à un moteur-fusée de grande puissance.

3.2.2 Les phases de vol :

le vol de la fusée se décompose en trois phases :

- La phase propulsée

- La phase balistique
- La descente sous parachute

Ces phases s'articulent autour d'évènements majeurs : La période s'écoulant de l'instant de la mise à feu à la fin de combustion du propulseur, et qui s'appelle la phase propulsée. Elle comprend une partie où la fusée est guidée par la rampe de lancement et une partie où la fusée est livrée à elle-même.

Après l'extinction du propulseur commence la phase balistique pendant laquelle la fusée, uniquement soumise à son poids et à la résistance de l'air, exploite la vitesse acquise pendant la propulsion pour atteindre son altitude maximale.

Après la culmination, lorsque l'engin commence à retomber, la phase balistique se poursuit jusqu'à l'ouverture du parachute.

Bien sûr, on peut rencontrer des phases balistiques avortées lorsque le parachute s'ouvre avant la culmination, ou des vols balistiques complets sans ouverture de parachute moins.

Les trois phases représentées dans la figure suivante :

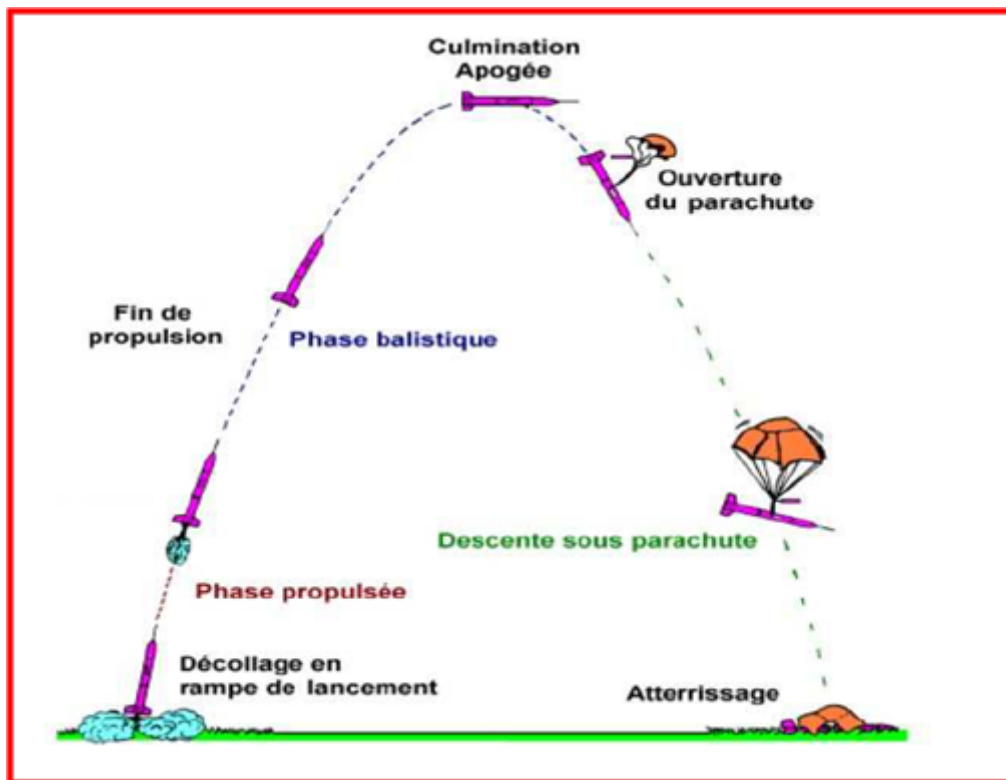


FIG. 3.1 – Illustration des phases d'un vol.

3.2.3 Les forces en présence :

Au cours de son vol, la fusée est soumise à trois forces :

- Le poids de la fusée,
- La poussée du moteur,
- La résistance de l'air à l'avancement de la fusée.

Le poids de la fusée :

Le poids s'exerce au Centre de Gravité de la fusée et est dirigé verticalement vers le bas. Si la fusée n'est pas verticale, on procède à la décomposition sur les axes de la fusée.

la poussée du moteur :

La poussée F s'applique au niveau du moteur, suivant l'axe longitudinal, vers l'ogive. En supposant que le propulseur est correctement positionné, la poussée s'applique au milieu de la plaque de poussée.

La résistance de l'air :

R s'oppose à l'avancement de la fusée dans l'air. Elle dépend donc du vent relatif, somme du vent crée par la vitesse de la fusée (vent vitesse) et du vent météo. Le vent relatif, ou "vent apparent", est le vent ressenti par la fusée.

Les trois forces représentées dans la figure suivante :

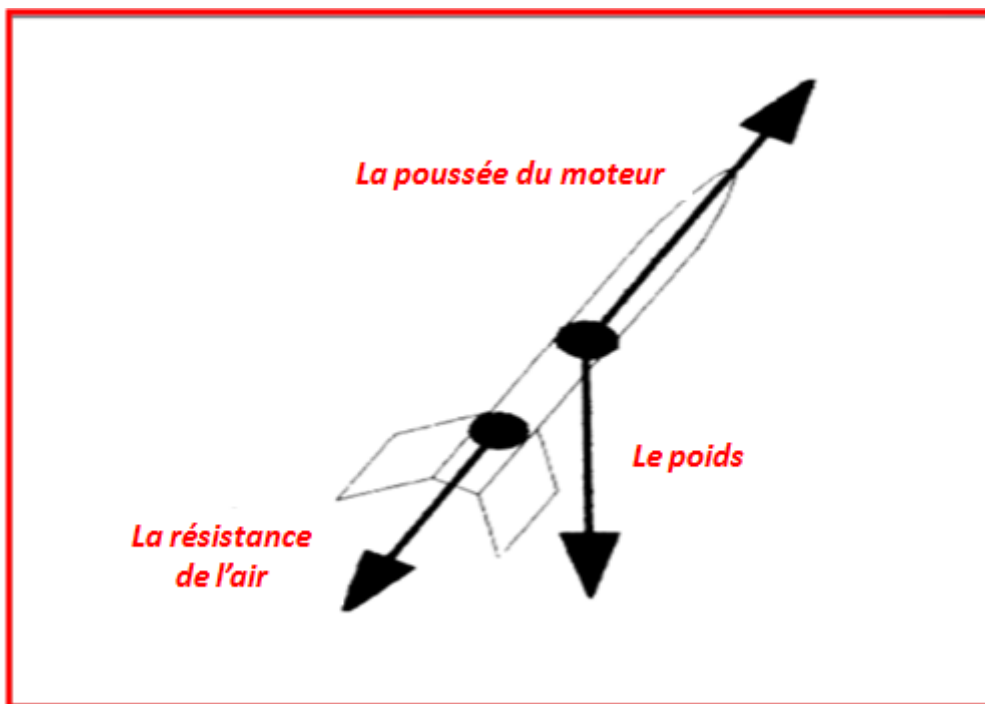


FIG 3.2 – Illustration des forces d'une fusée.

3.3 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 :

3.3.1 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)

$$\sum \vec{F} = 0$$

3.3.2 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 :

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

ou m : masse d'inertie de l'objet.

3.4 Position du problème :

Les deux chapitres précédents ont été consacrés à l'étude des équations différentielles et la théorie du contrôle optimal tandis que ce présent traite le le décollage d'une fusée dès son lancement jusqu'à atteindre son apogée.

Notre objectif dans cette partie est de minimiser l'énergie finale de cette fusée à ascendance verticale qui est soumise pendant son vol à l'influence de la pression atmosphérique et le carré inverse du champ gravitationnel.

Notons par :

$h(t)$: l'altitude au temps t .

$v(t)$: $t \in [0, T]$ le module de la vitesse.

$m(t)$: la masse de la fusée au temps t .

$u(t)$: est le seule contrôle dans notre problème qui est la poussée appliquée à la fusée.

Équations du système considéré :

- D'après la loi de Newton, la vitesse est issue de la dérivée de la distance qui dans notre cas est la dérivée de la hauteur extrême de notre fusée, la contrainte sur la hauteur sera :

$$\dot{h}(t) = v(t), \quad \text{avec} \quad h(0) = 0 \quad (3,1)$$

- D'après le principe fondamental de la dynamique (deuxième loi de Newton), quand une force résultante s'exerce sur un objet, celui-ci est soumis à une accélération qui a la même direction que la force. La contrainte sur la vitesse sera donc :
la somme des forces est égale au produit de la masse m fois l'accélération γ .

$$\sum \vec{F} = m(t) \cdot \vec{\gamma}$$

$$\Updownarrow$$

$$u(t) - m(t)g - d(h(t), v(t)) = m(t) \cdot \gamma = m(t) \cdot \dot{v}(t)$$

$$\Updownarrow$$

$$\dot{v}(t) = \frac{u(t) - d(h(t), v(t))}{m(t)} - g, \quad v(0) = 0 \quad (3, 2)$$

tel que :

$d(h(t), v(t))$: la densité de l'air

g : la gravité terrestre donné par $\frac{1}{h^2}$ dû au changement d'altitude.

A noter que la masse de la fusée au temps t est égale à la masse de la fusée sans carburant plus la masse du carburant.

A chaque fois que le temps augmente, la masse $m(t)$ diminue et la poussée augmente, donc :

$$\dot{m}(t) = \frac{-u(t)}{c} \quad (3, 3)$$

Selon l'étude de Knut Graichen et Nicolas Petit [6] qui traite sur la maximisation de l'altitude parcourue par la fusée, la hauteur maximale atteinte est $h = 1.01271727$ en un temps $T = 0.20407108$ (valeurs normalisées) [6].

Le problème considéré dans ce cas est de minimiser l'énergie u en gardant les mêmes résultats trouvés dans l'étude de Knut Graichen et Nicolas Petit [6], La fonction objectif sera donc :

$$J(u(t)) = \int_0^T u^2 dt \rightarrow \min$$

Le problème de contrôle optimal obtenu de la modélisation est le suivant :

$$\left\{ \begin{array}{l} J(u(t)) = \int_0^T u^2 dt \rightarrow \min \\ \dot{h}(t) = v(t), \quad h(0) = 1 \\ \dot{v}(t) = \frac{u(t) - d(h(t), v(t))}{m(t)} - \frac{1}{h^2}, \quad v(0) = 0 \\ \dot{m}(t) = \frac{-u(t)}{c}, \quad m(0) = 1 \\ 0 \leq u(t) \leq 3.5 = u_{max}, \quad t \in [0, T] \\ 0.6 < m(t) < 1 \end{array} \right. \quad (3,4)$$

avec u_{max} : la poussée maximale .

A noter que dans ce modèle les états de l'altitude, de la vitesse, de la masse ainsi que le temps sont normalisés [6].

$d(h(t), v(t))$: la densité de l'air est une fonction qui varie de façon exponentielle avec l'altitude, elle est définie comme suit :

$$d(h(t), v(t)) = q(h(t), v(t)) \frac{C_d A}{M_0 g} \quad (3,5)$$

tel que $q(h(t), v(t))$ est la pression atmosphérique donnée par :

$$q(h(t), v(t)) = \frac{1}{2} p_0 v(t)^2 \exp(\beta(1 - h(t))) \quad (3,6)$$

on remplace (3,6) dans (3,5) on aura :

$$d(h(t), v(t)) = \frac{1}{2} p_0 v(t)^2 \exp(\beta(1 - h(t))) \frac{C_d A}{M_0 g} \quad (3,7)$$

alors le problème devient

$$\left\{ \begin{array}{l} J(u(t)) = \int_0^T u^2 dt \rightarrow \min \\ \dot{h}(t) = v(t), \quad h(0) = 1 \\ \dot{v}(t) = \frac{u(t) - \frac{1}{2} p_0 v(t)^2 \exp(\beta(1 - h(t))) \frac{C_d A}{M_0 g}}{m(t)} - \frac{1}{h^2}; \quad v(0) = 0 \\ \dot{m}(t) = \frac{-u(t)}{c}; \quad m(0) = 1 \\ 0 \leq u(t) \leq 3.5 = u_{max}, \quad t \in [0, T] \\ 0.6 < m(t) < 1 \end{array} \right. \quad (3.8)$$

avec $h(T) = 1.01271727$, $v(T) = 0$

les paramètres utilisées dans la modélisation sont données dans le tableau suivant :

paramètres	signification
β	taux de décroissance de la densité de l'air
c	vitesse de combustion du carburant
p_0	densité de l'air au niveau de la mer
C_d	coefficient de la pression atmosphérique
A	Surface frontale de contact
g	la gravité terrestre
M_0	la masse initiale

3.4.1 Résolution théorique (Principe du maximum)

notre problème initial :

$$\left\{ \begin{array}{ll} J(u(t)) = \int_0^T u^2 dt \rightarrow \min & \\ \dot{h}(t) = v(t), & h(0) = 1, \quad h(T) = 1.01271727 \\ \dot{v}(t) = \frac{u(t) - \frac{1}{2}p_0 v(t)^2 \exp(\beta(1-h(t))) \frac{C_d A}{M_0 g}}{m(t)} - \frac{1}{h^2} & v(0) = 0, \quad v(T) = 0 \\ \dot{m}(t) = \frac{-u(t)}{c}; & m(0) = 1 \\ 0 \leq u(t) \leq 3.5 = u_{max}, & t \in [0, T] \end{array} \right.$$

Le Hamiltonien du système s'écrit :

$$H(h, v, m, p, u) = -u^2 + p_h v + p_v \left(u - \frac{1}{2} v^2 \exp(\beta(1-h)) \frac{C_d A p_0}{M_0 g} - \frac{1}{h^2} \right) - p_m \frac{u}{c} \quad (3.9)$$

On maximise le hamiltonien H en utilisant le PMP on aura :

$$\max_{0 \leq u \leq 3.5} H = \max_{0 \leq u \leq 3.5} \left(-u^2 + p_h v + p_v \left[u - \frac{1}{2} v^2 \exp(\beta(1-h)) \frac{C_d A p_0}{M_0 g} - \frac{1}{h^2} \right] - p_m \frac{u}{c} \right) \quad (3.10)$$

on dérive le hamiltonien c'est à dire $\frac{\partial H}{\partial u}$

Ce qui est équivalent à maximiser :

$$\max_{0 \leq u \leq 3.5} \left(\frac{p_v}{m} - \frac{p_m}{c} - 2u \right)$$

Le contrôle est donc :

$$u^*(t) = \begin{cases} 0 & \text{si } \frac{p_v}{2m} - \frac{p_m}{2c} < 0 \\ u_{max} & \text{si } \frac{p_v}{2m} - \frac{p_m}{2c} > 3.5 \\ \frac{p_v}{2m} - \frac{p_m}{2c} & \text{sinon} \end{cases} \quad (3,8)$$

Les vecteurs adjoints :

$p_h(t)$, $p_v(t)$ et $p_m(t)$, $t \in [0, T]$, sont les composantes du vecteur adjoint elles sont telle que :

$$\begin{cases} \dot{p}_h(t) = -\frac{\partial H}{\partial h} = -p_v \left[\frac{v^2 \beta}{2m} \exp(\beta(1-h)) \frac{p_0 C_d A}{M_0 g} + \frac{2}{h^3} \right] \\ \dot{p}_v(t) = -\frac{\partial H}{\partial v} = \frac{v}{m} \exp(\beta(1-h)) \frac{p_0 C_d A}{M_0 g} p_v - p_h \\ \dot{p}_m(t) = -\frac{\partial H}{\partial m} = \left[u - \frac{1}{2} v^2 \exp(\beta(1-h)) \frac{p_0 C_d A}{M_0 g} \right] \frac{p_v}{m^2}. \end{cases} \quad (3.11)$$

Les conditions de transversalité :

$$\begin{cases} p_h(T) \in \mathbb{R} \\ p_v(T) \in \mathbb{R} \\ p_m(T) \in \mathbb{R} \end{cases} \quad (3.12)$$

Problème aux deux bouts :

$$(PDB) \left\{ \begin{array}{l} \dot{h}(t) = v(t) \\ \dot{v}(t) = \frac{u(t) - \frac{1}{2}v(t)^2 \exp(\beta(1-h(t)) \frac{p_0 C_d A}{M_0 g}}{m(t)} - \frac{1}{h^2} \\ \dot{m}(t) = \frac{-u(t)}{c} \\ \dot{p}_h(t) = -p_v(t)(v(t)^2 \frac{\beta}{2m} \exp(\beta(1-h(t)) \frac{p_0 C_d A}{M_0 g} + \frac{2}{h^3} \\ \dot{p}_v(t) = \frac{v(t)}{m(t)} \exp(\beta(1-h(t)) \frac{p_0 C_d A}{M_0 g} p_v(t) - p_h(t) \\ \dot{p}_m(t) = (u(t) - \frac{1}{2}v(t)^2 \exp(\beta(1-h(t)) \frac{p_0 C_d A}{M_0 g}) \frac{p_v(t)}{m(t)^2} \\ h(0) = 1, \quad h(T) = 1.01271727 \\ v(0) = 0, \quad v(T) = 0 \\ m(0) = 1, \quad m(T) \in \mathbb{R} \\ p_h(t) \in \mathbb{R}, \quad p_h(T) \in \mathbb{R} \\ p_v(t) \in \mathbb{R}, \quad p_h(T) \in \mathbb{R} \\ p_m(t) \in \mathbb{R}, \quad p_h(T) \in \mathbb{R} \end{array} \right. \quad (3.13)$$

avec

$$u^* = \begin{cases} 0 \text{ si} & \frac{p_v}{2m} - \frac{p_m}{2c} < 0 \\ u_{max} \text{ si} & \frac{p_v}{2m} - \frac{p_m}{2c} > 3.5 \\ \frac{p_v}{2m} - \frac{p_m}{2c} & \text{sinon} \end{cases} \quad (3.14)$$

en posant $y = (h; v; m; p_h; p_v; p_m) = (y_1; y_2; y_3; y_4; y_5; y_6)$

on a alors

$$\left\{ \begin{array}{l} \dot{y}_1 = y_2 \\ \dot{y}_2 = \frac{u - \frac{1}{2}y_2^2 \exp(1-y_1) \frac{p_0 C_d A}{M_0 g}}{y_3} - \frac{1}{y_1^2} \\ \dot{y}_3 = -\frac{u}{c} \\ \dot{y}_4 = -y_5 \left(\frac{\beta}{2y_3} y_2^2 \exp(1-y_1) \frac{p_0 C_d A}{M_0 g} + \frac{2}{y_1^3} \right) \\ \dot{y}_5 = -\frac{y_2}{y_3} \exp(\beta(1-y_1)) \frac{p_0 C_d A}{M_0 g} y_5 - y_4 \\ \dot{y}_6 = \left(u - \frac{1}{2}y_2^2 \exp(\beta(1-y_1)) p_0 C_d A \right) \frac{y_5}{y_3^2} \\ y_1(0) = 1 \quad y_1(T) = 1.01271727 \\ y_2(0) = 0 \quad y_2(T) = 0 \\ y_3(0) = 1 \quad y_3(T) \in \mathbb{R} \\ y_4(0) \in \mathbb{R} \quad y_4(T) \in \mathbb{R} \\ y_5(0) \in \mathbb{R} \quad y_5(T) \in \mathbb{R} \\ y_6(0) \in \mathbb{R} \quad y_6(T) \in \mathbb{R} \end{array} \right. \quad (3.15)$$

avec

$$u = \begin{cases} 0, & \text{si } \frac{y_5}{2y_3} - \frac{y_6}{2c} < 0 \\ u_{max}, & \text{si } \frac{y_5}{2y_3} - \frac{y_6}{2c} > 0 \\ \frac{y_5}{2y_3} - \frac{y_6}{2c} & \text{sinon} \end{cases} \quad (3.16)$$

donc le problème de deux bouts est équivalent à $\dot{Y} = F(t, Y)$, avec :

$$Y(0) = \begin{pmatrix} 1 \\ 0 \\ 1 \\ \mathbb{R} \\ \mathbb{R} \\ \mathbb{R} \end{pmatrix}, \quad Y(T) = \begin{pmatrix} 1.01271727 \\ 0 \\ \mathbb{R} \\ \mathbb{R} \\ \mathbb{R} \\ \mathbb{R} \end{pmatrix}$$

le problème de Cauchy associé au problème est :

$$\dot{y} = F(t, y), \quad y(0) = y^0 = \begin{pmatrix} 1 \\ 0 \\ 1 \\ y_4^0 \\ y_5^0 \\ y_6^0 \end{pmatrix}$$

la solution de notre système s'écrit $y(t, y^0)$, on aimerait qu'elle coïncide avec la fin de la trajectoire $y(T)$ de la manière suivante :

$$y(T, y^0) = y(T) = \begin{pmatrix} 1.01271727 \\ 0 \\ \mathbb{R} \\ \mathbb{R} \\ \mathbb{R} \\ \mathbb{R} \end{pmatrix}$$

la fonction de tir est définie par :

$$G(y^0) = y(T, y^0) - y(T)$$

résoudre le problème (PDB) est équivalent à trouver le zéro de la fonction de tir $G(y^0)$, c'est à dire chercher $y^0/G(y^0) = 0$

3.4.2 Résolution numérique

Résoudre le système d'équations différentielles revient à trouver le zéro de la fonction de tir $G(y^0)$ en utilisant la commande `fsolve` sous Matlab. Le programme suivant nous permet d'afficher les solutions du problème.

programme :

1. *function ti simple*
2. *clear all; clf; clc;*
3. *global x0 tf;*
4. *x0 = [1; 0; 1] ; tf = 0.20407108; P0 = [0; 0; 0];*
5. *options = optimset('display','iter','Algorithm','levenberg-marquardt')*
6. *[P0f, FVAL, EXITFLAG] = fsolve(@S, [P0; tf], options;*
7. *P0f, FVAL, EXITFLAG*
8. *options = odeset('AbsTol', 1e-9, 'RelTol', 1e-9);*
9. *[t, y] = ode45(@sys, [0; P0f(4)], [x0, P0f(1); P0f(2); P0f(3)], options;*
10. *subplot(421); plot(t, y(:, 1)); title('Trajectoire de h(t)'); grid on*
11. *subplot(422); plot(t, y(:, 2)); title('Trajectoire de v(t)'); grid on*
12. *subplot(423); plot(t, y(:, 3)); title('Trajectoire de m(t)'); grid on*

```

13. subplot(424); plot(t,y(:,4)); title('Trajectoirede p1(t)'); grid on
14. subplot(425); plot(t,y(:,5)); title('Trajectoirede p2(t)'); grid on
15. subplot(426); plot(t,y(:,6)); title('Trajectoirede p3(t)'); grid on
16. if (y(:,5)/2 * y(:,3) - (y(:,6)/2*c)) < 0;
17. y(:,7) = 0;
18. else if (y(:,5)/2 * y(:,3) - (y(:,6)/2 * c)) > 0;
19. y(:,7) = 3.5;
20. elsey(:,7) = (y(:,5)/2 * y(:,3) - (y(:,6)/2 * c));
21. end
22. end
23. subplot(427); plot(t,y(:,7)); title('Trajecoirede u(t)'); grid on
24. function Yzero = S(Y)
25. global x0 tf;
26. options = odeset('AbsT', 1e - 9, 'RelTol', 1e - 9);
27. [t,y] = ode45(@sys, [0;tf], [x0;Y], options);
28. Yzero = [y(end,4)
29. y(end,5)
30. y(end,6)];
31. function ydot = sys(t,y)
32. if (y(5)/2 * (y(3)) - (y(6)/2 * c)) < 0;
33. u = 0;
34. else
35. if ((y(5)/2 * (y(3)) - y(6)/2 * c)) > 3.5;
36. u = 3.5;
37. else
38. u = ((y(5)/2 * (y(3)) - y(6)/2 * c));
39. end
40. end
41. y1 = y2
42.  $y_2 = \frac{u - \frac{1}{2}y_2^2 \exp(1-y_1) \frac{p_0 C_d A}{M_0 g}}{y_3} - \frac{1}{y_1^2}$ 
43.  $y_3 = -\frac{u}{c}$ 
44.  $y_4 = -y_5 \left( \frac{\beta}{2y_3} y_2^2 \exp(1-y_1) \frac{p_0 C_d A}{M_0 g} + \frac{2}{y_1^2} \right)$ 
45.  $y_5 = -\frac{y_2}{y_3} \exp(\beta(1-y_1)) \frac{p_0 C_d A}{M_0 g} y_5 - y_4$ 
46.  $y_6 = (u - \frac{1}{2}y_2^2 \exp(\beta(1-y_1))) p_0 C_d A \frac{y_5}{y_3^2}$ 
47. end

```

La méthode étant sensible à l'initialisation c'est à dire aux choix des conditions initiales, nous avons pas eu les résultats souhaités, ce qui va nous induire à utiliser la méthode directe.

programme

```
1. function directe
2. %descritisation en utilisant fmicon
3. clear all; close all; clc;
4. global tf e B l;
5. e = 0.5; B = 500; l = 620;
6. Umax = 3.5;
7. n = 200; tf = 0.20407108;
8. uinit = Umax * rand(n,1); % initialisation aléatoire du controle
9. xinit = uinit ; % point de départ pour fmincon
10. lb = zeros(n,1); ub = Umax * ones(n,1); %contraintes sur le controle 0 <=
    u < Umax
11. options = optimset('Display','iter','Algorithm','active-set');
12. [res,Fval,exitflag] = fmincon(@fobj,xinit,[],[],[],[],lb,ub,@cond,options);
13. exitflag
14. % ===== Calcul des trajectoires optimales =====
15. x(1) = 1; z(1) = 1; y(1) = 0;
16. for i = 1 : n - 1
17. k11 = (tf/n) * y(i);
18. k21 = (tf/n) * (y(i) + k11/2);
19. k31 = (tf/n) * (y(i) + k21/2);
20. k41 = (tf/n) * (y(i) + k31);
21. x(i + 1) = x(i) + (k11 + 2 * k21 + 2 * k31 + k41)/6;
22. k12 = (tf/n) * (((res(i) - 1/2 * y(i)^2 * exp(B * (1 - x(i)) * 1))/z(i)) - 1/x(i)^2);
23. k22 = (tf/n) * (((res(i) - 1/2 * y(i)^2 * exp(B * (1 - x(i)) * 1))/z(i)) - 1/x(i)^2 + k12/2);
24. k32 = (tf/n) * (((res(i) - 1/2 * y(i)^2 * exp(B * (1 - x(i)) * 1))/z(i)) - 1/x(i)^2 + k22/2);
25. k42 = (tf/n) * (((res(i) - 1/2 * y(i)^2 * exp(B * (1 - x(i)) * 1))/z(i)) - 1/x(i)^2 + k32);
26. y(i + 1) = y(i) + (k12 + 2 * k22 + 2 * k32 + k42)/6;
27. k13 = (tf/n) * (-res(i)/e);
28. k23 = (tf/n) * (-res(i)/e + k13/2);
29. k33 = (tf/n) * (-res(i)/e + k23/2);
30. k43 = (tf/n) * (-res(i)/e + k33/2);
31. z(i + 1) = z(i) + (k13 + 2 * k23 + 2 * k33 + k43)/6;
32. end
33. end
34. % ===== tracé des courbes =====
35. subplot(221); plot(linspace(0,tf,n),x(1:n),'b'); title('Hauteur'); grid on
36. subplot(221); plot(linspace(0,tf,n),y(1:n),'b'); title('vitesse'); grid on
37. subplot(221); plot(linspace(0,tf,n),z(1:n),'b'); title('Masse'); grid on
38. subplot(221); plot(linspace(0,tf,n),res(1:n),'b'); title('Contrôle'); grid on
```

```

39. % ===== contrainte du probleme =====
40. function [c,ceq] = cond(u)
41. global tf e b l;
42. n = length(u)
43. xf = 1; yf = 0; zf = 1;
44. for i = 1 : n
45. k11 = (tf/n) * yf;
46. k21 = (tf/n) * (yf + k11/2);
47. k31 = (tf/n) * (yf + k21/2);
48. k41 = (tf/n) * (yf + k31);
49. xf = xf + (k11 + 2 * k21 + 2 * k31 + k41)/6;
50. k12 = (tf/n) * (((u(i) - 1/2 * xyf * exp(B * (1 - xf)) * 1)/z(i)) - 1/xf^2);
51. k22 = (tf/n) * (((u(i) - 1/2 * yf^2 * exp(B * (1 - xf)) * 1)/z(i)) - 1/xf^2 + k12/2);
52. k32 = (tf/n) * (((u(i) - 1/2 * yf^2 * exp(B * (1 - xf)) * 1)/z(i)) - 1/xf^2 + k22/2);
53. k42 = (tf/n) * (((u(i) - 1/2 * yf^2 * exp(B * (1 - xf)) * 1)/z(i)) - 1/xf^2 + k32);
54. yf = yf + (k12 + 2 * k22 + 2 * k32 + k42)/6;
55. k13 = (tf/n) * (-u(i)/e);
56. k23 = (tf/n) * (-u(i)/e + k13/2);
57. k33 = (tf/n) * (-u(i)/e + k23/2);
58. k43 = (tf/n) * (-u(i)/e + k33/2);
59. zf = zf + (k13 + 2 * k23 + 2 * k33 + k43)/6;
60. c = [c; abs(zf(i) - 0.8) - 0.2];
61. end
62. ceq = [xf - 1.01271727, yf];
63. % ===== declaration de la fonction objectif =====
64. fonction val = fobj(u)
65. global tf;
66. n = length(u);
67. val = 0;
68. for i = 1 : n
69. k11 = (tf/n) * u(i)^2;
70. k21 = (tf/n) * (u(i)^2 + k11/2);
71. k31 = (tf/n) * (u(i)^2 + k21/2);
72. k41 = (tf/n) * (u(i)^2 + k31);
73. val = val + (k11 + 2 * k21 + 2 * k31 + k41)/6;
74. end

```


les Résultats d'exécution sont donnés dans la figure suivante :

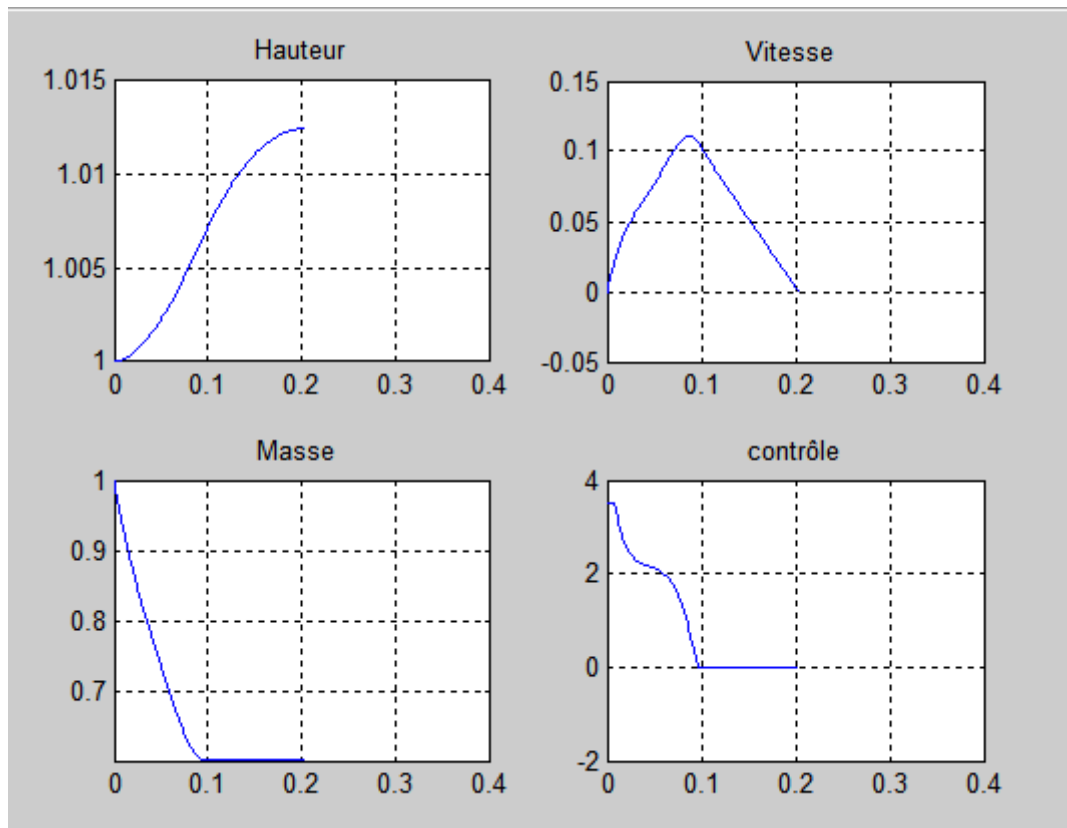


FIG 3.3 Résultats par la méthode directe

Les résultats obtenues par Knut Graichen et Nicolas Petit [6] sont donnés dans la figure suivante :

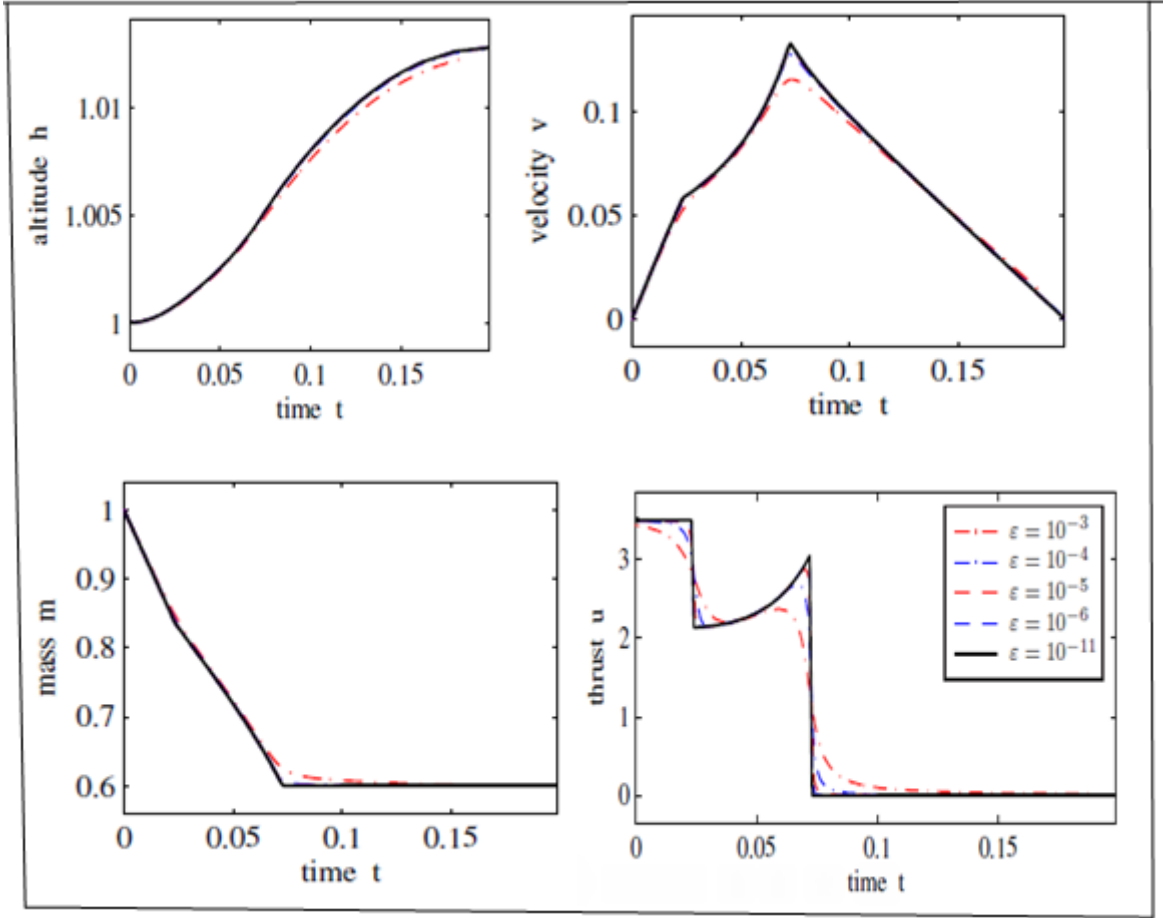


FIG 3.4 Résultats obtenues par Knut Graichen et Nicolas Petit

3.5 Discussion des résultats

La figure 3.3 nous donne les mêmes cibles obtenues dans l'étude de Knut Graichen et Nicolas Petit, c'est à dire notre fusée atteint l'altitude maximale $h = 1.01271727$ au bout de $t = 0.20407108$ (altitude et temps normalisés) c'est a dire $81km$ au bout de 12.4 minutes et $m(t_f) = 0.6$ (tout le carburant est épuisé). La différence réside dans les structures du contrôle.

Notre problème qui est la minimisation de l'énergie est équivalent au problème de maximisation de la masse. En effet

$$\max m(t_f) \iff \min -m(t_f)$$

$$\iff \min -m(t_f) + m(0) - m(0)$$

$$\iff \min -\int_0^{t_f} \dot{m}(t) dt - m(0)$$

$$\iff \min \int \frac{u(t)}{c} dt - m(0)$$

$$\Longleftrightarrow \min \int_0^{t_f} u^2(t) dt$$

car $u(t) > 0$.

Le problème de maximisation de la masse et de maximisation de l'altitude donnent des courbes aux allures équivalentes pour l'altitude, la vitesse, et la masse sauf pour la courbe du contrôle.

En effet, supposons que le problème de maximisation de la masse donne une valeur $m(t_f) > 0.6$. Ceci signifie qu'il reste du carburant dans le réservoir, ce qui implique qu'avec cette énergie qui reste on peut encore aller plus haut, ce qui contredit le fait que l'altitude est maximale.

Si on veut atteindre une altitude de 60 km c'est à dire $h(t_f) = 1.00975$ alors la simulation donne les courbes suivantes :

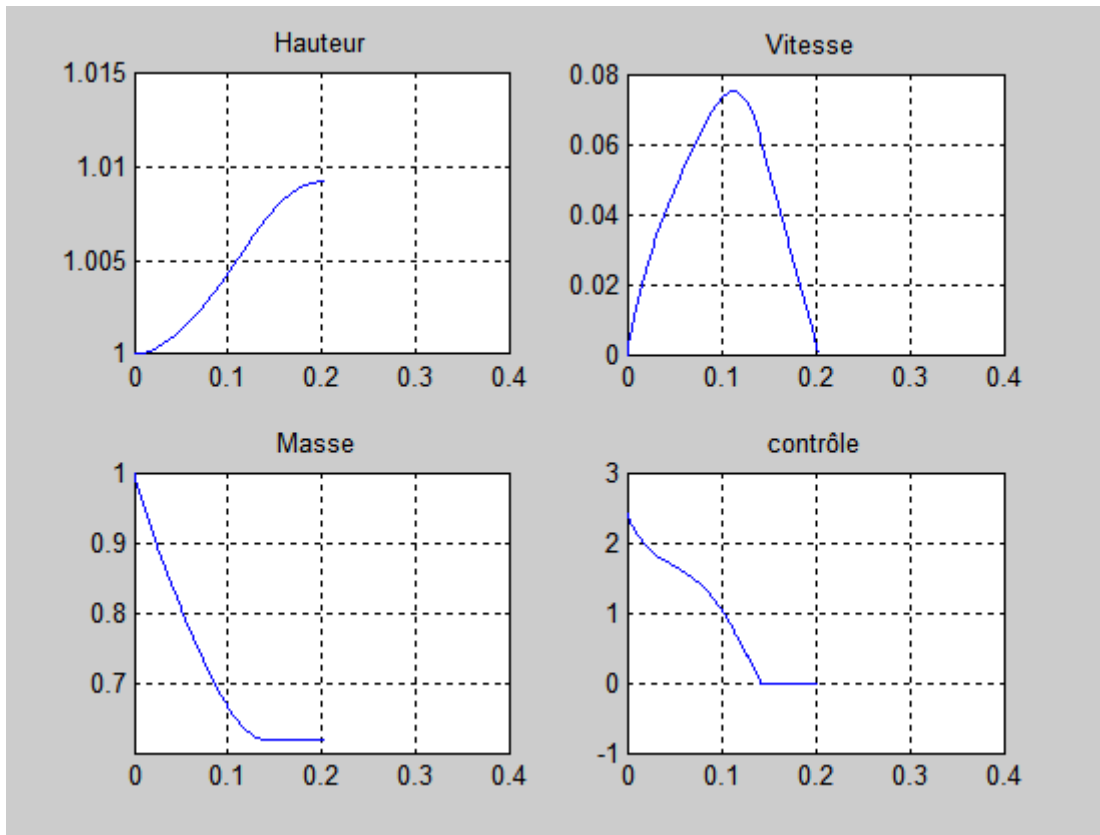


FIG 3.5-Résultats par la méthode directe.

La masse finale $m(t_f) = 0.62$, c'est à dire que si le réservoir ne contient que 40% de la masse alors les 2% qui restent peut être remplacés par une masse utile pour la fusée.

Conclusion générale

Notre objectif dans ce mémoire est l'étude d'un problème de contrôle optimale dont on veut minimiser l'énergie, Nous avons modélisé le problème de GODDARD (décollage d'une fusée) en tenant compte de toute les forces externes y compris la pression atmosphérique et le pesanteur.

La modélisation d'un système de contrôle optimale a recours a des équations différentielles, c'est pourquoi on a entamé notre travail par une généralités sur les équations différentielles. Ensuite une présentation de la théorie du contrôle optimal est réalisée à savoir ; la contrôlabilité des systèmes linéaires et non linéaires, quelques méthodes de résolution. En particulier on s'est intéressé à la méthode du principe du maximum de Pontryagin.

Par la suite on a programmé sous MATLAB deux méthodes de résolution à savoir la méthode indirecte appelée méthode de tir, puis la méthode directe appelée aussi (tir directe), nous avons commencé par la méthode indirecte ou nous avons appliqué le principe de maximum de pontriaguin qui nous a donné les conditions d'optimalité, On a remarqué que la méthode indirecte est très sensible au choix du point de départ, c'est pour cela nous sommes passé à la résolution par la méthode directe en discrétisant le problème via la technique de Runge-Kutta.

Bibliographie

- [1] J.P. Demailly. Analyse numérique et équations différentielles. Université de Grenoble. EDP Sciences., (2006).
- [2] E. Trélat. Contrôle optimal : Théorie et Applications. Université Paris-Sud. Laboratoire EN-EDP. Mathématique UMR 8628 Bat 425 ; 91405 Orsay Cedex, (2005).
- [3] Boudi.Z. Le contrôle optimal en finance. Thèse de Master. Université Mouloud Mammeri Tizi-ouzou.
- [4] R.E. Kalman. Mathematical discription of linear dynamical system. siam. journal on control., 1963.
- [5] P.Tsiotrast et H. J. Kelley. Drag-law Effects in the Goddard Problem. Automatica, Vol. 27, No. 3, pp. 481-490, 1991 Printed in Great Britain.
- [6] Knut Graichen Nicolas Petit. Solving the Goddard problem with thrust and dynamic pressure constraints using saturation functions. Proceedings of the 17th World Congress The International Federation of Automatic Control Seoul, Korea, July 6-11, 2008.
- [7] P.Tsiotrast et H. J. Kelley. Goddard Problem with Constrained Time of Flight. Virginia Polytechnic Institute and State University, Blacksburg, Virginia 24061. JOURNAL OF GUIDANCE, CONTROL, AND DYNAMICS Vol. 15, No. 2, March-April 1992
- [8] Steven Finch . Goddard's Rocket Problem. June 12, 2015. (cours).
- [9] Hans Seywald et Clifft . Virginia Polytechnic Institute and State University, Blacksburg, Virginia. JOURNAL OF GUIDANCE, CONTROL, AND DYNAMICS Vol. 16, No. 4, July-August 1993.