

Numéro d'ordre : 3429

# Thèse

présentée à

## L'UNIVERSITÉ BORDEAUX I

École Doctorale des  
Sciences Physiques et de l'Ingénieur

PAR CHRISTOPHE LOUEMBET

pour obtenir le grade de

**DOCTEUR**

SPÉCIALITÉ : AUTOMATIQUE, PRODUCTIQUE, SIGNAL ET IMAGE

---

### GENERATION DE TRAJECTOIRES OPTIMALES POUR SYSTEMES DIFFERENTIELLEMENT PLATS

### APPLICATION AUX MANOEUVRES D'ATTITUDE SUR ORBITE

---

Soutenue le 21 septembre 2007

Après avis de :

Mme	BRIGITTE D'ANDRÉA NOVEL	Professeur, Ecole des Mines de Paris
et M.	DENIS ARZELIER	Chargé de Recherche HDR, LAAS-CNRS, Toulouse

Devant la commission d'examen formée de :

Mme	BRIGITTE D'ANDRÉA NOVEL	Professeur, Ecole des Mines de Paris
M.	DENIS ARZELIER	Chargé de Recherche HDR, LAAS-CNRS, Toulouse
M.	FRANCK CAZAURANG	Maître de conférences, Université Bordeaux 1
Mme	CATHERINE CHARBONNEL	Thales Alenia Space, Cannes
M.	THIERRY DARGENT	Thales Alenia Space, Cannes
M.	ALAIN OUSTALOUP	Professeur, ENSEIRB
Mme	CHRISTELLE PITTET-MECHIN	Le CNES, Toulouse
M.	ALI ZOLGHADRI	Professeur, Université Bordeaux 1

– 2007 –



A mon père



# Remerciements

Je tiens à remercier avant tout, les membres du jury qui me font l'honneur de participer à l'examen de ce travail.

Je tiens notamment à remercier Madame Brigitte D'ANDRÉA-NOVEL de l'école des Mines de Paris et Monsieur Denis ARZELIER du LAAS de Toulouse pour avoir accepté d'être rapporteurs pour ma thèse.

J'exprime ma gratitude au Centre National d'Etudes Spatiales et à l'entreprise THALES ALENIA SPACE pour avoir financé les travaux présentés dans ce mémoire.

Je tiens à exprimer mes plus sincères remerciements à mon Directeur de Thèse Ali ZOLGHADRI pour m'avoir accueilli au sein de son équipe de recherche. Je remercie aussi toutes celles et ceux qui ont participé à l'encadrement de cette thèse et contribué à la qualité scientifique de ce document : Catherine CHARBONNEL de THALES ALENIA SPACE, Christelle PITTET-MECHIN du CNES et Frank CAZAURANG de l'IMS de Bordeaux.

Un grand merci à toute l'équipe ARIA : Jérôme (Trois ans assis sur une pierre l'un en face de l'autre sans avoir réussi à les réchauffer), Sylvain (On a toujours un truc à dire), Madjid (On l'a fait malgré les difficultés). Je n'oublie pas David (encore merci pour tes précieux conseils), Tarek (Je suis plus qu'un quota), Loic (Le landais du Pays Basque), Vincent (On travaille ensemble quand tu veux), Alexandre (Ne fais pas cette bêtise!), Gaetan (Que Linux soit avec toi!). Vous avez tous participé à cette ambiance de travail qui rend le réveil moins difficile.

Un grand merci à toutes celles et ceux qui m'ont soutenu. Tout d'abord à ma chère et aimée Sylvie avec qui je partage mes plus grands moments de bonheur. Elle m'a (sup)porté durant toute les phases difficiles de la vie de thésard. De plus, si ce document est de qualité c'est en grande partie grâce à elle. Alors j'espère que je serai à la hauteur lorsque tu auras besoin de moi. Je pense à ma famille (au sens africain) de Lille à Pointe-Noire en passant par Paris, Belley, Toulouse et Lomé (Il n'y a pas d'arbre fort sans racines profondes). Un merci tout particulier à ma mère à qui je dois mon envie de réussir à tout prix et bien plus encore (Tu imagines! Tous les sacrifices auxquels tu as consenti pour que ton fils devienne un jour docteur!). Encore merci à tous mes amis parisiens, Guillaume dit le clown, Caroline, Emmanuelle, Patrick et tout le BDE Roots et mes amis bordelais Dominique (We are young, black and gifted), Benjamin, Sophie et Matsouma.

A toutes les rencontres qui m'ont amené jusqu'ici : je pense à David Brochot (Le principal est

d'être un leader), Christophe Talbot (Merci de m'avoir donné ma chance), Denis Arzelier, Didier Henrion, Dimitri Peaucelle (Le trio du (...de) MAC (LAAS) : au fond c'est parce que je voulais vous ressembler que j'en suis là) et Ali Zolghadri (Tu m'a toujours poussé à exprimer le meilleur de moi-même).

La dernière phrase de ces remerciements sera à la mémoire de mon père qui, j'en suis certain, veille sur moi et aurait été très heureux d'assister à ma soutenance.

# Table des matières

<b>Remerciements</b>	<b>iii</b>
<b>Liste des figures</b>	<b>x</b>
<b>Liste des tableaux</b>	<b>xi</b>
<b>Notations</b>	<b>xiii</b>
<b>Introduction générale</b>	<b>1</b>
<b>I De la génération de trajectoires optimales d'attitude</b>	<b>5</b>
I.1 Introduction . . . . .	6
I.2 Position du problème de guidage . . . . .	8
I.3 Problème de commande optimale : formulation indirecte . . . . .	9
I.3.1 Conditions nécessaires d'optimalité . . . . .	10
I.3.2 Méthodes de tirs . . . . .	15
I.3.3 Méthodes indirectes de transcription . . . . .	18
I.4 Problème de commande optimale : formulation directe . . . . .	18
I.4.1 Méthodes directes de tirs . . . . .	19
I.4.2 Méthodes directes de transcription : collocation . . . . .	20
I.5 Méthodes composites et autres méthodes . . . . .	23
I.5.1 Association des méthodes de collocation et des méthodes indirectes de tirs multiples . . . . .	23
I.5.2 Autres méthodes de génération de trajectoires . . . . .	23
I.6 Conclusion . . . . .	23
<b>II Platitude différentielle et génération de trajectoires</b>	<b>25</b>
II.1 Introduction . . . . .	26
II.2 Jets infinis et équivalence de Lie-Bäcklund . . . . .	26
II.2.1 Jets infinis . . . . .	27
II.2.2 Système piloté . . . . .	29
II.2.3 Equivalence de Lie-Bäcklund . . . . .	31
II.3 Condition nécessaire et suffisante de platitude différentielle . . . . .	32
II.3.1 Platitude Différentielle . . . . .	32
II.3.2 Condition nécessaire et suffisante . . . . .	34
II.4 Platitude du modèle dynamique d'attitude d'un satellite rigide . . . . .	40
II.4.1 Modèle dynamique d'attitude d'un satellite rigide . . . . .	40

II.4.2	Applications des conditions nécessaires et suffisantes . . . . .	44
II.4.3	Sorties plates et bouclage dynamique . . . . .	45
II.5	Apports de la platitude au problème de commande optimale . . . . .	46
II.6	Conclusion . . . . .	47
<b>III</b>	<b>Génération de trajectoires optimales par platitude et collocation</b>	<b>49</b>
III.1	Introduction . . . . .	51
III.2	Contexte . . . . .	51
III.3	Optimisation paramétrique globale . . . . .	52
III.3.1	Rappel du problème de commande optimale plat . . . . .	52
III.3.2	Paramétrisation des sorties plates par B-splines . . . . .	54
III.3.3	Problème d'optimisation paramétrique . . . . .	55
III.4	Optimisation paramétrique sous-optimale . . . . .	56
III.4.1	Approximation convexe intérieure . . . . .	57
III.4.2	Problème d'optimisation convexe . . . . .	63
III.4.3	Mesure du conservatisme . . . . .	65
III.5	Etude d'une trajectoire . . . . .	70
III.5.1	Définition de la mission . . . . .	70
III.5.2	Etude des paramètres de configuration . . . . .	71
III.5.3	Optimisation non convexe . . . . .	73
III.5.4	Optimisation convexe . . . . .	76
III.5.5	Comparaison des différents algorithmes . . . . .	80
III.6	Conclusions . . . . .	83
<b>IV</b>	<b>Mise en oeuvre : application sur simulateur</b>	<b>85</b>
IV.1	Introduction . . . . .	86
IV.2	Générateur de trajectoires MANIAC . . . . .	86
IV.3	Simulations . . . . .	87
IV.3.1	Description du simulateur . . . . .	87
IV.3.2	Environnement et conditions de simulations . . . . .	89
IV.3.3	Solutions calculées par platitude et collocation . . . . .	89
IV.3.4	Discussion et comparaison des résultats de simulation . . . . .	89
IV.4	Conclusions . . . . .	92
<b>V</b>	<b>Génération de trajectoires : formulation semi-définie positive</b>	<b>95</b>
V.1	Introduction . . . . .	96
V.2	Représentation en somme de carrés . . . . .	96
V.2.1	Cas des ordres impairs : $k_v = 2k_u - 1$ . . . . .	97
V.2.2	Cas des ordres pairs : $k_v = 2k_u$ . . . . .	101
V.3	Application au problème de génération de trajectoires . . . . .	102
V.3.1	Appartenance d'une trajectoire $t \mapsto \bar{z}(t)$ à un demi-espace de $O_{\bar{z}}$ . . . . .	102
V.3.2	Appartenance d'une trajectoire $t \mapsto \bar{z}(t)$ à l'intersection de plusieurs demi- espaces de $O_{\bar{z}}$ . . . . .	110
V.3.3	Application au problème de guidage . . . . .	113
V.4	Conclusions . . . . .	114

---

<b>Conclusions générale</b>	<b>117</b>
<b>Références bibliographiques</b>	<b>119</b>
<b>Références bibliographiques</b>	<b>121</b>
<b>Annexes</b>	<b>126</b>
<b>A Programmation non linéaire</b>	<b>127</b>
A.1 Problème d'optimisation non contraint . . . . .	127
A.2 Contraintes égalitaires . . . . .	128
A.3 Contraintes inégalitaires . . . . .	129
<b>B Eléments de géométrie différentielle</b>	<b>131</b>
B.1 Morphismes . . . . .	131
B.2 Théorèmes des fonctions implicites . . . . .	132
B.3 Variété différentielle . . . . .	132
B.4 Fibré tangent, Champ de vecteurs et Dérivée de Lie . . . . .	133
B.5 Fibré cotangent et champ de 1-formes . . . . .	134
<b>C Espaces de fonctions polynomiales par morceaux et B-splines</b>	<b>135</b>
C.1 Espaces de fonctions polynomiales par morceaux et B-splines . . . . .	135
<b>D Superellipsoïdes : définitions complémentaires</b>	<b>139</b>
D.1 Volume d'une superellipsoïde . . . . .	139
<b>E Analyse par intervalles et inversion ensembliste</b>	<b>143</b>
E.1 Analyse par intervalles . . . . .	143
E.2 Inversion ensembliste par arithmétique d'intervalles . . . . .	145



# Liste des figures

I.1	Classification des méthodes de résolution numérique du problème de commande optimale . . . . .	7
I.2	Discrimination des arcs sur une trajectoire contrainte . . . . .	13
I.3	Approximation de la trajectoire d'Hermite-Simpson sur un segment . . . . .	21
II.1	Véhicule non holonome roulant sans glisser . . . . .	39
II.2	PMR par projection Stéréographique des quaternions d'attitude . . . . .	42
III.1	Schéma de configuration du problème d'optimisation . . . . .	56
III.2	Variation de la superellipse en fonction de $\varepsilon$ . . . . .	59
III.3	Vecteur définissant une superellipsoïde comme produit sphérique de deux superellipses . . . . .	60
III.4	Superellipsoïdes 3D pour différentes valeurs de $\varepsilon_1$ et $\varepsilon_2$ . . . . .	61
III.5	Projection des pavés $\mathbf{z}_i$ suivant $z_1$ et $z_2$ . . . . .	66
III.6	Projection des pavés $\mathbf{z}, \dot{\mathbf{z}}_i$ suivant $z_1$ et $\dot{z}_1$ . . . . .	67
III.7	Projection des pavés $\mathbf{z}, \dot{\mathbf{z}}, \ddot{\mathbf{z}}_i$ suivant $z_1$ et $\ddot{z}_1$ . . . . .	68
III.8	Profils de trajectoire pour une base de B-splines d'ordre 6 . . . . .	72
III.9	Profils de trajectoire pour une base de B-splines d'ordre 6 . . . . .	72
III.10	Profils de trajectoire pour une base de B-splines d'ordre 7 et 10 points de collocation distribués uniformément . . . . .	73
III.11	Profils de trajectoire pour une base de B-splines d'ordre 6 et 10 points de collocation avec une distribution adaptée . . . . .	74
III.12	Processus de recherche du temps optimal . . . . .	75
III.13	Temps optimal de transfert . . . . .	75
III.14	Nombre d'itérations pour la procédure non convexe . . . . .	76
III.15	Temps d'exécution pour la procédure non convexe . . . . .	76
III.16	Temps optimal de transfert pour une approximation ellipsoïdale . . . . .	77
III.17	Nombre d'itérations sous contraintes quadratiques . . . . .	78
III.18	Temps d'exécution sous contraintes quadratiques . . . . .	78
III.19	Profils de trajectoire obtenus sous contraintes quadratiques . . . . .	79
III.20	Temps minimal de transfert sous contraintes linéaires . . . . .	79
III.21	Nombres d'itérations sous contraintes linéaires . . . . .	80
III.22	Temps d'exécution sous contraintes linéaires . . . . .	80
III.23	Temps minimal de transfert sous approximation superellipsoïdale $\mathcal{S}_1$ . . . . .	81
III.24	Nombre d'itérations sous contraintes superellipsoïdales . . . . .	81
III.25	Temps d'exécution sous contraintes superellipsoïdales . . . . .	82

III.26	Temps minimal de transfert sous approximation superellipsoïdale $\mathcal{S}_2$ . . . . .	82
III.27	Temps minimal de transfert sous approximation superellipsoïdale $\mathcal{S}_4$ . . . . .	83
III.28	Temps minimal de transfert sous approximation superellipsoïdale $\mathcal{S}_5$ . . . . .	83
IV.1	Profils de vitesse calculés par MANIAC . . . . .	87
IV.2	Boucle SCAO mode normal (* roues à réaction) . . . . .	88
IV.3	Temps de transfert optimal pour le satellite DEMETER en fonction de la base de B-splines . . . . .	90
IV.4	Profils de vitesse et de couple calculés par platitude pour une base de B-spline d'ordre 6 . . . . .	90
IV.5	Trajectoire d'attitude simulée . . . . .	91
IV.6	Ecart par rapport à la trajectoire d'attitude nominale . . . . .	91
IV.7	Vitesses angulaires du satellite . . . . .	92
IV.8	Vitesse de rotation des roues à réaction . . . . .	93
IV.9	Couple de sorties des roues à réaction . . . . .	93
V.1	Trajectoire pour une base de B-spline d'ordre 7 et 8 sous contrainte linéaire . . . . .	111
V.2	Trajectoire pour une base de B-spline d'ordre 7 et 8 sous contrainte d'appartenance à un polytope . . . . .	112
V.3	Trajectoire pour une base de B-spline d'ordre 8 sous contrainte linéaire . . . . .	114
E.1	Effets d'enveloppement pour $\theta = -\pi/4$ . . . . .	147

# Liste des tableaux

III.1	Calcul des superellipsoïdes . . . . .	63
III.2	Comparaison des algorithmes . . . . .	84
IV.1	Ecart maximum à la trajectoire nominale . . . . .	92
IV.2	Consommation de couple en phase de tranquillisation . . . . .	92
E.1	Algorithme SIVIA . . . . .	146



# Notations

$Q$	quaternion normalisé d'attitude
$q_0$	partie réel du quaternion normalisé
$q_1$	partie imaginaire du quaternion normalisé
$q_2$	partie imaginaire du quaternion normalisé
$q_3$	partie imaginaire du quaternion normalisé
$\Psi$	vecteur des angles de Cardan
$\sigma$	vecteur des paramètres modifiés de Rodrigues
$\Omega$	vecteur vitesse de rotation
$p$	première composante de $\Omega$
$q$	seconde composante de $\Omega$
$r$	troisième composante de $\Omega$
$\mathbb{I}_{PF}$	matrice d'inertie du satellite
$H_{tot}$	vecteur du moment cinétique totale du satellite
$H_{PF}$	vecteur du moment cinétique de la plateforme
$H_{int}$	vecteur du moment cinétique des roues à réaction dit moment interne
$\dot{H}_{int}$	vecteur du couple développé par les roues à réaction
$z$	vecteur des sorties plates
$\bar{z}$	vecteur des sorties plates et ses dérivées successives
$O_{\bar{z}}$	Espace des sorties plates et de ses dérivés
$S_{\bar{z}}$	sous espace de $O_{\bar{z}}$ admissible vis à vis des contraintes
$S_n$	superellipsoïde de dimension $n$
$\mathcal{E}_n$	ellipsoïde de dimension $n$
$\theta$	vecteur des anomalies
$\varepsilon$	vecteur des exposants trigonométrique définissant $S_n$
$\mathcal{F}_n$	fonction d'appartenance à $S_n$
$\mathcal{P}_n$	polytope en dimension $n$
$\mathbb{P}_{k,\xi}$	Espace des polynômes par morceaux d'ordre $k$ défini sur la séquence $\xi$
$\mathbb{P}_{k,\xi,v}$	Espace des polynômes par morceaux d'ordre $k$ défini sur la séquence $\xi$ respectant la séquence de continuité $v$
$\xi$	séquence de segmentation
$T$	séquence nodale temporelle
$v_i$	degré de continuité en $\xi_i$
$\rho_i$	multiplicité de $\xi_i$ dans la séquence $T$
$\mathcal{H}$	Hamiltonien du problème à deux bouts
$\mathcal{L}$	Lagrangien du problème à deux bouts



# Introduction générale

Depuis que la concurrence internationale s'est accrue dans le secteur spatial, la réalisation de nouvelles missions a dû intégrer le principe de diminution des coûts sans pour autant faire des concessions sur les performances et la fiabilité des satellites. Dans cette optique, le Centre National d'Etudes Spatiales (CNES) et THALES ALENIA SPACE ont souhaité se doter d'outils de génération de trajectoires d'attitude adaptés aux spécificités de leurs plateformes satellitaires, MYRIADE et PROTEUS. Ces plateformes ont été les deux premières à être produites en série pour les missions scientifiques d'observation de la Terre. Elles ont été conçues pour être adaptables et reconfigurables afin de réduire le coût de conception de chaque mission.

La génération de trajectoires correspond, dans le domaine du contrôle d'attitude sur orbite, au terme de guidage. Elle consiste à définir les manoeuvres en rotation autour du centre de gravité du véhicule ; en d'autres termes, déterminer les profils temporels de l'état et de la commande associée qui respectent un ensemble de contraintes. La problématique de la génération de trajectoires d'attitude est une composante essentielle du Système de Contrôle d'Attitude sur Orbite (SCAO) dans le contexte des satellites manoeuvrant d'observation de la Terre. En effet, à l'exception du mode de survie, l'ensemble des modes du SCAO comporte des manoeuvres en attitude, à savoir :

- **Mode grossier de transition** : il consiste en une manoeuvre d'attitude permettant de basculer de la position acquise durant le mode survie vers une attitude proche de celle du mode normal.
- **Mode normal** : Il permet d'assurer l'ensemble des pointages et d'effectuer toutes les manoeuvres d'évitement d'éblouissement des senseurs.
- **Mode de contrôle d'orbite** : Il consiste à basculer le satellite de manière à orienter les tuyères pour le contrôle d'orbite.

La génération de trajectoires d'attitude a fait l'objet de nombreux travaux depuis les années 70, notamment aux Etats Unis (cf. [Scrivener & Thomson 1994] et les références incluses) et en Russie (cf. [Levskii 2002] et les références incluses). Dans ces travaux, le problème est généralement envisagé comme un problème de commande optimale qui ne possède pas de solution analytique connue, bien que sa formulation soit relativement facile et immédiate [Scrivener & Thomson 1994]. De manière générale, il existe deux méthodes classiques pour résoudre le problème de commande optimale [Betts 1998] : les méthodes dites indirectes qui se basent sur l'utilisation du principe du maximum de Pontryagin [Pontryagin, Boltyanskii, Gamkrelidze & Mishchenko 1962] et les méthodes dites directes qui formulent le problème en termes de programmation paramétrique non linéaire. Les applications des méthodes indirectes au domaine du guidage ont été largement étudiées. Mentionnons notamment [Li & Bainum 1990], [Bilimoria & Wie 1993] et [Levskii 2002]. Les applications des méthodes directes au domaine du guidage, comme les travaux de Scrivener [Scrivener & Thomson 1993], ont néanmoins été plus rares.

Lorsque l'on cherche à minimiser le temps de manoeuvre, les méthodes évoquées précédemment mènent généralement à considérer des profils de couples saturées de type Bang-Bang. Ces profils discontinus offrent l'intérêt d'être particulièrement adaptés lorsque les actionneurs sont des propulseurs à ergols. Or, les plateformes, telles que MYRIADE ou PROTEUS, sont pilotées par des roues à réaction qu'il faut éviter de saturer. La prise en compte de cette contrainte nécessite la mise en place de méthodes de génération de trajectoires adaptées. De plus, il est important de considérer l'impact des profils de couples utilisés sur les structures souples du véhicule. Il paraît donc important de pouvoir définir et générer des profils de couples suffisamment continus afin d'éviter les phénomènes d'excitation des modes souples. Pour répondre à ces différentes attentes, nous utilisons une approche alternative de la génération de trajectoires fondée sur le concept de platitude différentielle [Fliess, Lévine, Martin & Rouchon 1995]. La platitude différentielle est une propriété structurelle d'une classe de systèmes dynamiques, où il est possible d'exprimer toutes les variables du système (état et commande) en fonction d'une variable spécifique (appelée sortie plate ou linéarisante) et de ses dérivées. Cette propriété permet de caractériser toutes les contraintes de génération de trajectoires par des contraintes sur les sorties plates et leurs dérivées. La platitude différentielle constitue ainsi une réelle avancée conceptuelle dans la résolution du problème de commande optimale.

Parce qu'elles suppriment l'intégration durant le processus d'optimisation, les propriétés de la platitude différentielle permettent de reformuler le problème de commande optimale en un problème d'optimisation *géométrique*. Le problème consiste dès lors à obtenir une trajectoire de l'espace reliant deux points tout en étant inscrite dans un volume défini. Du fait des propriétés différentielles des sorties plates, plusieurs méthodes de résolution s'offrent alors à nous. Dans [van Nieuwstad 1997], l'auteur définit une méthode pour la génération de trajectoires par platitude sans contraintes sur la trajectoire. Dans [Lévine & Ngyuen 2003], une interpolation polynomiale est utilisée pour définir les trajectoires de sorties plates, et la durée de la trajectoire en relâchée en présence de contraintes. Dans [Milam, Mushambi & Murray 2000], les auteurs introduisent l'utilisation de la collocation, conjointement à la platitude, pour développer une méthodologie de génération de trajectoires sous contraintes. Les outils de collocation sont classiques dans la résolution du problème de commande optimale. En association avec la platitude, ces outils permettent de formuler le problème de commande optimale en un problème d'optimisation paramétrique non linéaire. Enfin, Faiz [Faiz, Agrawal & Murray 2001] propose une méthode permettant de définir un cadre de contraintes linéaires pour la résolution du problème commande optimale par platitude.

Les travaux présentés dans ce mémoire se situent dans le cadre général des méthodes de génération de trajectoires utilisant le concept de platitude différentielle. Au plan méthodologique, notre contribution propre s'articule selon trois axes. Le premier axe, situé en amont du problème de génération de trajectoire, concerne la caractérisation de la platitude du modèle dynamique d'attitude et la détermination des sorties plates par l'application des conditions nécessaires et suffisantes de Lévine [Lévine 2004b]. Le second axe, placé dans la lignée des travaux de Milam et Faiz [Milam 2003, Faiz 1999], vise à appliquer les méthodes de résolution du problème de commande optimale par platitude et collocation dans un cadre de contraintes convexes. Enfin, dans le dernier axe de recherche, nous nous proposons de reformuler le problème de commande optimale en termes de problème d'optimisation semi-définie.

Le mémoire est structuré de la façon suivante :

Le **chapitre I** est consacré à un état de l'art sur la génération de trajectoires optimales par résolution du problème de commande optimale. Nous aborderons les méthodes de résolution directes et indirectes. Nous analyserons leur utilisation dans le cadre de la génération de trajectoires optimales d'attitude.

Le **chapitre II** aborde la notion de platitude différentielle et l'implication de ses propriétés structurelles dans le problème de commande optimale. La contribution de ce chapitre porte sur l'application de la condition nécessaire et suffisante au modèle de la dynamique d'attitude d'un satellite rigide et dans la détermination des sorties plates.

Le **chapitre III** se situe dans le cadre de génération de trajectoires d'attitude par platitude différentielle et collocation par B-splines. Notre contribution concerne ici le développement de techniques de convexification du problème d'optimisation paramétrique non linéaire obtenu par la platitude et la collocation. La convexification vise à améliorer les performances numériques de l'algorithme de génération de trajectoires.

Le **chapitre IV** est entièrement consacré à la mise en oeuvre des techniques développées dans le chapitre III sur un simulateur à caractère industriel (interne au CNES). Il s'agit, pour un scénario donné, de comparer les profils générés par platitude et collocation et les profils générés par les techniques actuellement employées au CNES. L'analyse des résultats obtenus permet de mettre en évidence les avantages des techniques basées sur la platitude différentielle.

Enfin, dans le **chapitre V**, nous développons une méthode novatrice dont le principal avantage est de prendre en compte les contraintes de manière continue en temps. Elle permet de s'affranchir du problème de vérification des contraintes *a posteriori*, intrinsèque aux techniques de collocation. Cette méthode vise à formuler le problème de génération de trajectoires de sorties plates sous contraintes linéaires comme la conjonction de problèmes de positivité de polynômes par morceaux. La caractérisation de la positivité d'un polynôme par morceaux se traduit comme un problème de résolution d'une Inégalité Matricielle Linéaire ( $\mathcal{LMI}$ ). La génération de trajectoires optimales d'attitude se pose alors comme un problème d'optimisation  $\mathcal{LMI}$ , qui peut être résolu par des techniques dédiées.



---

# Chapitre I

## De la génération de trajectoires optimales d'attitude

---

### Sommaire

---

<b>I.1</b>	<b>Introduction</b>	<b>6</b>
<b>I.2</b>	<b>Position du problème de guidage</b>	<b>8</b>
<b>I.3</b>	<b>Problème de commande optimale : formulation indirecte</b>	<b>9</b>
I.3.1	Conditions nécessaires d'optimalité	10
I.3.1.1	Problème aux deux bouts non contraint	10
I.3.1.2	Contraintes égalitaires	11
I.3.1.3	Contraintes inégalitaires	12
I.3.1.4	Application au guidage	14
I.3.2	Méthodes de tirs	15
I.3.2.1	Méthodes indirectes de tirs	15
I.3.2.2	Méthodes indirectes de tirs multiples	16
I.3.3	Méthodes indirectes de transcription	18
<b>I.4</b>	<b>Problème de commande optimale : formulation directe</b>	<b>18</b>
I.4.1	Méthodes directes de tirs	19
I.4.2	Méthodes directes de transcription : collocation	20
<b>I.5</b>	<b>Méthodes composites et autres méthodes</b>	<b>23</b>
I.5.1	Association des méthodes de collocation et des méthodes indirectes de tirs multiples	23
I.5.2	Autres méthodes de génération de trajectoires	23
<b>I.6</b>	<b>Conclusion</b>	<b>23</b>

---

## I.1 Introduction

Ce premier chapitre aborde la problématique du guidage en attitude des satellites. L'objectif est de formuler le problème de guidage comme un problème de commande optimale et de présenter les solutions classiques existant dans la littérature. Ce tour d'horizon devrait permettre au lecteur d'aborder les développements méthodologiques des chapitres suivants. Le niveau de détails utilisé dans ce chapitre retranscrit la richesse et l'abondance des travaux sur la résolution des problèmes de commande optimale, domaine qui s'est fortement développé pour répondre aux problèmes posés dans le cadre de l'exploration spatiale. Nous nous efforcerons de présenter les principaux outils de résolution, de clarifier nos propos par l'application au domaine du guidage et de souligner quelques problèmes ouverts.

Nous supposons par la suite que le comportement dynamique du système considéré est décrit par un modèle d'état non linéaire sous la forme :

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

où  $x(t)$  représente le profil temporel de l'état du système et  $u(t)$  celui de la commande.

### Définition I.1 (Problème de commande optimale)

Le problème de commande optimale consiste à déterminer une commande  $u(t)$  qui permet à la fois :

- de vérifier les conditions initiales et finales sur les profils temporels de l'état et de la commande :

$$\Psi(x(t_0), u(t_0), t_0, x(t_f), u(t_f), t_f) = 0, \quad \Psi \in \mathbb{R}^{n_\Psi}$$

- de satisfaire des contraintes imposées au système qui peuvent être :
  - soit de nature instantanée :  $\forall t \in [t_0, t_f], c_t(x(t), u(t)) \geq 0, \quad c_t \in \mathbb{R}^{n_{c_t}}$
  - soit de nature intégrale :  $\forall t \in [t_0, t_f], \int_{t_0}^t c_i(x(\tau), u(\tau)) d\tau \geq 0, \quad c_i \in \mathbb{R}^{n_{c_i}}$
- d'optimiser un critère de coût

$$J = \int_{t_0}^{t_f} \phi(x(t), u(t)) dt, \quad \phi \in \mathbb{R} \quad (1.2)$$

où  $\phi$  est une fonction de coût.

La théorie de la commande optimale dans le cadre d'un système non linéaire sous contraintes est un domaine actif de recherche dont on peut trouver un état de l'art dans [Sussmann & Willems 1997]. Afin d'en faciliter la mise en oeuvre, des méthodes numériques fiables et performantes (en termes de convergence, temps de calcul et degré de conservatisme) ont été développées [Lastman 1978], [Bryson 1999], [Stoer & Bulirsch 1980], [Hull 1997]. Ces méthodologies associent une méthode de résolution du problème de commande optimale à une technique d'intégration numérique.

Il existe principalement deux grands courants méthodologiques de résolution du problème de commande optimale. Les méthodes indirectes utilisent les conditions nécessaires d'optimalité issues du principe du maximum de Pontryagin et transforment le problème de commande optimale en un *problème aux deux bouts*. En effet, le comportement de la commande étant défini par les conditions d'optimalité, l'optimisation porte sur les conditions initiales des trajectoires

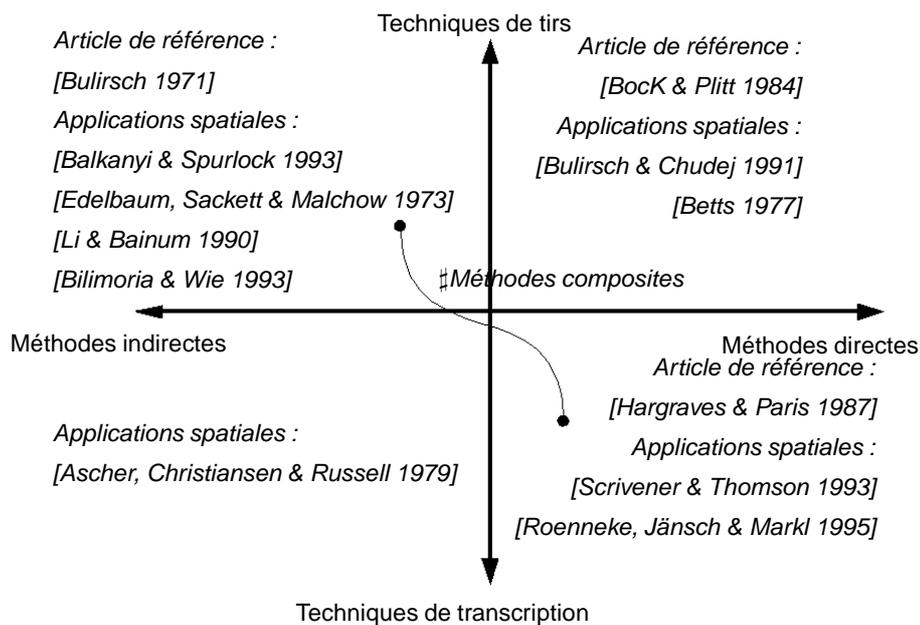


FIG. I.1 – Classification des méthodes de résolution numérique du problème de commande optimale

des variables associées au problème aux deux bouts de manière à rallier les conditions finales désirées. Les méthodes directes transforment le problème de commande optimale en un problème d'optimisation paramétrique. Pour cela, l'évolution des commandes et/ou des états du système est décrite à l'aide de courbes paramétrées. L'optimisation porte alors sur les paramètres de ces courbes qui doivent vérifier les conditions initiales et finales ainsi que les contraintes sur la trajectoire. Les conditions d'optimalité sont assurées par le processus d'optimisation non linéaire utilisé.

Pour chacune des deux approches citées ci-dessus, la dynamique du système doit être vérifiée. Il existe deux techniques communément associées à la résolution du problème de commande optimale [Betts 1998] : des techniques de *tirs* ou de *transcription*. Dans le cas des techniques de tirs, les équations différentielles sont intégrées explicitement de manière numérique. Les méthodes de transcription consistent à vérifier la dynamique du système en un nombre d'instant finis grâce à un jeu de contraintes supplémentaires sur les variables d'optimisation. La figure I.1 donne plusieurs démarches utilisant les méthodes de résolution du problème de commande optimale et les techniques d'intégration citées précédemment dans le cadre de la génération de trajectoires pour les systèmes spatiaux. Les références données dans [Scrivener & Thomson 1994] et [Shen & Tsiotras 1999] fournissent un panorama des méthodes classiques utilisées pour le guidage en attitude.

La suite de ce chapitre est organisée de la façon suivante. Nous formulerons tout d'abord le problème de guidage comme un problème de commande optimale. Puis, nous présenterons les méthodes indirectes et directes de résolution du problème de commande optimale. Enfin, nous évoquerons les méthodes composites qui sont basées sur les points forts des méthodes directes et indirectes.

## I.2 Position du problème de guidage

Commençons par décrire le modèle d'état non linéaire d'un satellite piloté par des roues à réaction. Le satellite rigide est défini par sa matrice d'inertie supposée pleine. Le modèle mathématique d'un tel système est composé des équations de la cinématique des paramètres d'attitude et des équations de la dynamique en rotation. Les paramètres d'attitude permettent de définir la rotation entre le repère inertiel  $R_i$  et le repère véhicule  $R_v$ . La représentation d'attitude la plus commune dans la littérature est la représentation par les quaternions normalisés. Un quaternion normalisé  $Q$  est associé à la rotation principale  $\mathcal{R}$  d'angle  $\theta$  autour d'un vecteur unitaire  $e = [e_1 e_2 e_3]^T$  :

$$Q = \begin{pmatrix} \cos \frac{\theta}{2} \\ \sin \frac{\theta}{2} e \end{pmatrix} = \begin{pmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{pmatrix} \quad (1.3)$$

La rotation  $\mathcal{R}$  est unique si l'on impose :  $0 \leq \theta \leq \pi$ . En notant le vecteur vitesse de rotation entre  $R_i$  et  $R_v$  par  $\Omega_{R_v/R_i} = \Omega = [p, q, r]^T$ , la propagation des quaternions d'attitude est décrite dans [Dulot 2003] par :

$$\dot{Q} = \frac{1}{2} Q \Omega \quad (1.4)$$

Sachant qu'un produit de deux quaternions se ramène à une multiplication matricielle, il vient alors :

$$\begin{bmatrix} \dot{q}_0 \\ \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \end{bmatrix} = \frac{1}{2} \begin{bmatrix} q_0 & -q_1 & -q_2 & -q_3 \\ q_1 & q_0 & -q_3 & q_2 \\ q_2 & q_3 & q_0 & -q_1 \\ q_3 & -q_2 & q_1 & -q_0 \end{bmatrix} \begin{bmatrix} 0 \\ p \\ q \\ r \end{bmatrix} \equiv \frac{1}{2} Q \Omega \quad (1.5)$$

Les équations de la dynamique sont issues du principe fondamental de la dynamique (2<sup>ème</sup> loi de Newton) :

$$\left( \frac{dH_{tot}}{dt} \right)_{/R_i} = \sum \text{couples extérieurs} \quad (1.6)$$

où  $H_{tot}$  représente le moment cinétique total du véhicule (dans le cas d'un corps rigide  $H = \mathbb{I} \cdot \Omega$ ). La dérivée temporelle de  $H_{tot}$  se décompose dans les coordonnées du repère véhicule  $R_v$  de la manière suivante :

$$\left( \frac{dH_{tot}}{dt} \right)_{/R_i} = \left( \frac{dH_{tot}}{dt} \right)_{/R_v} + \Omega \wedge H_{tot} \quad (1.7)$$

En remarquant que le système considéré comporte le satellite et les roues à réaction, nous rappelons les deux points suivants

- Les seuls couples qui s'exercent sur le système sont des couples internes à celui-ci puisqu'ils proviennent de l'accélération (positive ou négative) des roues à réaction du véhicule. Ainsi, en l'absence de couples extérieurs, nous avons conservation du moment cinétique :

$$H_{tot/R_i} = \text{constante} \quad (1.8)$$

**Remarque I.1** *Le modèle que nous considérons dans ce document s'inscrit dans le cadre de la génération de trajectoire en boucle ouverte. Ainsi, toutes les forces perturbatrices*

exogènes et non mesurables ne sont pas modélisées. Nous supposons que la boucle de régulation les rejette conformément à un cahier des charges en pilotage.

– Le moment cinétique du système se décompose comme suit :

$$H_{tot} = H_{plateforme} + H_{int} \quad (1.9)$$

où  $H_{plateforme} = \mathbb{I}_{PF} \cdot \Omega$  et  $H_{int} = \mathbb{I}_{roue} \cdot \Omega_{roue}$ .

Ainsi, nous pouvons écrire à partir de l'équation (1.7) l'équation dynamique intrinsèque du satellite :

$$\mathbb{I}_{PF} \dot{\Omega} + \dot{H}_{int} + \Omega \wedge H_{tot} = 0 \quad (1.10)$$

Les équations de la dynamique d'un satellite rigide se résument alors :

$$\dot{Q} = \frac{1}{2} Q \Omega \quad (1.11)$$

$$\mathbb{I}_{PF} \dot{\Omega} = -\dot{H}_{int} - \Omega \wedge H_{tot} \quad (1.12)$$

On notera les états et les entrées de commande par :

$$x = [Q^T, \Omega^T]^T = [q_0, q_1, q_2, q_3, p, q, r]^T \quad (1.13)$$

$$u = \dot{H}_{int}^T = [\dot{H}_{int,1}, \dot{H}_{int,2}, \dot{H}_{int,3}]^T \quad (1.14)$$

Il vient alors

$$\dot{x} = f(x, u) = \begin{pmatrix} \frac{1}{2} Q \Omega \\ \mathbb{I}_{PF}^{-1} (-\dot{H}_{int} - \Omega \wedge H_{tot}) \end{pmatrix} \quad (1.15)$$

Nous définissons le problème de génération de trajectoires comme un problème de commande optimale. Suivant le but recherché, la fonction de coût  $\phi(x(t), u(t))$  peut être définie par le temps de manoeuvre ou encore par la consommation du couple. Les états devront aussi respecter les conditions aux extrémités de la trajectoire à savoir :

$$\Psi(x(t_0), u(t_0), x(t_f), u(t_f)) = \begin{pmatrix} x(0) - x_0 \\ x(t_f) - x_f \\ u(0) - u_0 \\ u(t_f) - u_f \end{pmatrix} = 0 \quad (1.16)$$

Les entrées de commande devront respecter les contraintes liées aux capacités des actionneurs en vitesse ( $H_{int,max}$ ) et en accélération ( $\dot{H}_{int,max}$ ) :

$$c(x(t), u(t)) = \begin{pmatrix} \dot{H}_{int,max} - |u| \\ H_{int,max} - \left| \int_0^t u(\tau) d\tau \right| \end{pmatrix} \geq 0 \quad (1.17)$$

Le problème de guidage étant posé, présentons maintenant les méthodes indirectes de résolution du problème de commande optimale.

### I.3 Problème de commande optimale : formulation indirecte

Après avoir posé des conditions nécessaires d'optimalité issues du principe du maximum de Pontryagin pour un système contraint, nous présenterons les différentes techniques d'intégration numérique associées à la formulation indirecte du problème de commande optimale. Le

qualificatif indirecte vient du fait que nous allons nous aider de variables supplémentaires, appelées multiplicateurs de Lagrange, dont certains possèdent une dynamique. L'obtention d'une solution au problème passe alors par la détermination de ces multiplicateurs de Lagrange et, le cas échéant, de leurs trajectoires.

### I.3.1 Conditions nécessaires d'optimalité : application du principe du maximum de Pontryagin

Dans cette partie, nous nous limiterons à rappeler les principales notions du principe du maximum de Pontryagin. Ainsi, nous donnerons les conditions d'optimalité pour le problème initial (problème aux deux bouts). Puis, nous introduirons les conditions d'optimalité lorsque que le système est contraint.

#### I.3.1.1 Problème aux deux bouts non contraint

##### Définition I.2 (Problème aux deux bouts)

On appelle problème aux deux bouts non contraint, le problème qui consiste à définir les entrées de commandes  $u(t)$  minimisant le critère  $J$  :

$$J = \int_{t_0}^{t_f} \phi(x(t), u(t)) dt, \quad \phi \in \mathbb{R} \quad (1.18)$$

en respectant, d'une part, l'équation d'état du système :

$$\dot{x} = f(x(t), u(t)) \quad (1.19)$$

et d'autre part, les conditions aux limites de la trajectoire :

$$\Psi(x(t_0), u(t_0), t_0, x(t_f), u(t_f), t_f) = 0, \quad \Psi \in \mathbb{R}^{n_\Psi} \quad (1.20)$$

Soit  $\mathcal{H}$  l'Hamiltonien associé au problème précédent, défini par :

$$\mathcal{H}(x, u, t) = \lambda^T(t) f(x(t), u(t)) \quad (1.21)$$

et la fonction auxiliaire :

$$\Phi = \phi + v^T \Psi \quad (1.22)$$

$v \in \mathbb{R}^{n_\Psi}$  et  $\lambda(t) \in \mathbb{R}^n$  sont appelés des multiplicateurs de Lagrange. Les variables composant le vecteur  $\lambda(t)$  sont aussi connu comme les *états adjoints* ou *co-états* du problème d'optimisation. Les dérivées partielles de l'Hamiltonien et de la fonction auxiliaire par rapport à  $x$  et  $u$  sont définies de la façon suivante :

$$\mathcal{H}_x = \left( \frac{\partial \mathcal{H}}{\partial x_1}, \dots, \frac{\partial \mathcal{H}}{\partial x_n} \right) \quad (1.23)$$

$$\mathcal{H}_u = \left( \frac{\partial \mathcal{H}}{\partial u_1}, \dots, \frac{\partial \mathcal{H}}{\partial u_m} \right) \quad (1.24)$$

$$\Phi_x = \left( \frac{\partial \Phi}{\partial x_1}, \dots, \frac{\partial \Phi}{\partial x_n} \right) \quad (1.25)$$

Pontryagin [Pontryagin et al. 1962] a montré que trouver une solution optimale au sens du critère  $J$  respectant (I.19) et (I.20) équivaut à trouver une trajectoire  $t \mapsto (x(t), u(t), \lambda(t))$  telle que l'Hamiltonien associé  $\mathcal{H}$  soit minimal vis-à-vis de toutes les entrées de commande. Il énonce alors le principe du maximum :

$$u = \min_u \mathcal{H} \quad (I.26)$$

Les conditions nécessaires d'optimalité d'Euler-Lagrange s'obtiennent à partir de la condition d'extremum sur  $J$  à savoir  $dJ = 0$  et du principe du maximum de Pontryagin (I.26). Elles sont formées par :

les équations adjointes,

$$\dot{\lambda} = -\mathcal{H}_x^T \quad (I.27)$$

les équations de contrôle,

$$\mathcal{H}_u^T = 0 \quad (I.28)$$

et les conditions transverses,

$$\begin{cases} \lambda(t_0) = \Phi_x^T|_{t_0} \\ (\dot{\Phi} + \mathcal{H})|_{t_0} = 0 \\ \lambda(t_f) = \Phi_x^T|_{t_f} \\ (\dot{\Phi} + \mathcal{H})|_{t_f} = 0 \end{cases} \quad (I.29)$$

Les équations de contrôle (I.28) sont l'application directe du principe du maximum de Pontryagin.

La résolution du problème aux deux bouts nécessite non seulement l'intégration de la dynamique du système (I.19) dont la commande est déterminée par l'équation (I.28), mais aussi l'intégration de la dynamique des états adjoints (I.27). Or, si les conditions initiales et finales sont généralement connues pour les variables d'état, elles ne le sont généralement pas pour les états adjoints. Nous nous trouvons donc face à un problème aux limites. Le principal point dur de la méthode est de définir des outils qui permettent d'obtenir des valeurs convenables pour les conditions initiales (voir paragraphe I.3.2.1).

Notons de plus que les conditions d'optimalité (I.27), (I.28), (I.29) ne sont que des conditions nécessaires. En effet, il n'y a pas unicité de la trajectoire répondant aux conditions d'optimalité.

Nous venons de montrer que le problème aux deux bouts peut être résolu par l'application de conditions nécessaires d'optimalité dérivées du principe du maximum de Pontryagin. Montrons maintenant qu'il est possible de prendre en compte des contraintes égalitaires et inégalitaires.

### I.3.1.2 Contraintes égalitaires

Considérons un problème de commande optimale présenté dans la définition I.2 auquel on rajoute les contraintes égalitaires définies de la façon suivante :

$$c(x(t), u(t)) = 0, \quad c \in \mathbb{R}^l \quad (I.30)$$

Définissons la matrice jacobienne liée à l'ensemble de contraintes  $c(x(t), u(t))$  :

$$C_u = \nabla_u c(x(t), u(t)) = \begin{bmatrix} \frac{\partial c_1}{\partial u_1} & \frac{\partial c_1}{\partial u_2} & \cdots & \frac{\partial c_1}{\partial u_m} \\ \frac{\partial c_2}{\partial u_1} & \frac{\partial c_2}{\partial u_2} & \cdots & \frac{\partial c_2}{\partial u_m} \\ \vdots & & \ddots & \\ \frac{\partial c_l}{\partial u_1} & \frac{\partial c_l}{\partial u_2} & \cdots & \frac{\partial c_l}{\partial u_m} \end{bmatrix}, \quad (1.31)$$

Si la matrice  $C_u$  est de rang plein i.e.  $\forall t_i \in [t_0, t_f]$ ,  $\text{Rang}(C(t_i)) = \min(l, m)$ , alors nous pouvons redéfinir l'Hamiltonien tel que :

$$\mathcal{H}_a = \lambda^T(t) f(x(t), u(t)) + \mu^T c(x(t), u(t)) \quad (1.32)$$

avec  $\mu \in \mathbb{R}^l$ . Les conditions nécessaires d'optimalité (1.27), (1.28) s'en trouvent modifiées en conséquence.

Il peut arriver que la matrice  $C_u$  ne soit pas de rang plein. Dans ce cas, il est possible de réécrire un jeu de contraintes équivalentes  $c'(x(t), u(t), t)$  à partir de la dérivation temporelle des contraintes  $c(x(t), u(t), t)$ . On vérifie alors le rang de la matrice  $C'_u$ . Si cette matrice est de plein rang, on remplace les contraintes  $c(x(t), u(t), t)$  par  $c'(x(t), u(t), t)$ . Dans le cas contraire, on réitère l'opération. Cette approche est connue sous le nom de réduction d'ordre dans le domaine des équations algèbre-différentielles [Ascher, Mattheij & R. 1988, Brenan, Campbell & Petzold 1989].

Comme nous venons de le voir, dans le cas d'une trajectoire contrainte, les conditions d'optimalités (1.27), (1.28) et (1.29) se trouvent modifiées par la substitution de  $\mathcal{H}$  par  $\mathcal{H}_a$ . Il faut ainsi discerner les arcs de trajectoires où les contraintes sont satisfaites des arcs où elles ne le sont pas, ce qui nous amène à diviser la trajectoire en arcs non contraints et en arcs contraints. Les arcs non contraints seront traités selon les conditions d'optimalité du paragraphe 1.3.1.1, et les arcs contraints selon les conditions d'optimalité (1.27), (1.28) et (1.29) modifiées par l'Hamiltonien augmenté (1.32).

### 1.3.1.3 Contraintes inégalitaires

Considérons un problème de commande optimale présenté dans la définition 1.2 sous les contraintes inégalitaires suivantes

$$c(x(t), u(t)) \geq 0, \quad c \in \mathbb{R}^l \quad (1.33)$$

Après un premier calcul de la trajectoire qui ne prend pas en compte les contraintes (i.e. la trajectoire initiale), les contraintes inégalitaires sont séparées en deux sous ensembles : les contraintes enfreintes  $c \leq 0$  dites *actives* et les contraintes respectées  $c > 0$  dites *inactives*. Seules les contraintes actives sont considérées. La trajectoire est divisée en arcs sur lesquels une liste des contraintes actives est actualisée. Si, sur un arc, la liste des contraintes actives est vide, on dit que l'arc est non contraint et le problème sur cet arc est alors résolu comme dans le paragraphe 1.3.1.1. Si la liste n'est pas vide, on dit que l'arc est contraint alors le problème sur cet arc est résolu en utilisant l'approche proposée au paragraphe 1.3.1.2.

Pour illustrer la discrimination des arcs, prenons l'exemple d'une trajectoire scalaire  $t \mapsto x(t)$  d'un système soumis aux contraintes suivantes :

$$c(x(t), u(t)) = \begin{pmatrix} c_1 - x(t) \\ c_2 - x(t) \end{pmatrix}$$

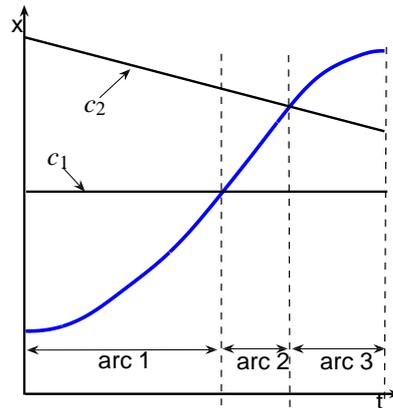


FIG. I.2 – Discrimination des arcs sur une trajectoire contrainte

suivant la figure I.2. Ainsi, l'arc 1 de la trajectoire est libre de toute contrainte et les conditions d'optimalité sont alors données par les équations (I.27), (I.28) et (I.29). L'arc 2 est contraint puisque la condition  $c_1$  est active sur celui-ci. De même, les conditions  $c_1$  et  $c_2$  sont actives sur l'arc 3, il est ainsi contraint. Pour les arcs 2 et 3, le problème est résolu en imposant les contraintes égalitaires

$$\begin{pmatrix} c_1 - x(t) \\ c_2 - x(t) \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

La prise en compte de contraintes inégalitaires entraîne plusieurs difficultés :

- La commande  $u$  se définit à partir de la condition (I.28) si la matrice hessienne  $\mathcal{H}_{u,u} = \nabla_u^2 \mathcal{H}$  est non singulière. Dans le cas où  $\mathcal{H}_{u,u}$  est singulière, alors la commande  $u$  n'est pas définie de manière unique et l'arc est alors qualifié de singulier. La résolution de tels problèmes met en jeu des techniques similaires à celles utilisées dans le cas où  $C_u$ , la matrice jacobienne des contraintes actives, est singulière.
- Le nombre et le séquençement des différents arcs ainsi que les instants de jonction entre les différents arcs ne sont pas connus *a priori*.
- Il se peut qu'il y ait discontinuité de la commande  $u$  et des co-états  $\lambda$  aux jonctions entre les arcs. Lorsque l'on a une connaissance *a priori* des arcs, il peut s'avérer pratique de transformer ce problème en plusieurs sous problèmes aux deux bouts correspondant aux divers arcs tout en imposant des contraintes transverses supplémentaires pour assurer la continuité des variables [Pesch 1994, Bryson & Ho 1975].

Ainsi, la théorie de la commande optimale apporte des conditions nécessaires d'optimalité bien établies d'un point de vue formel. Cependant, plusieurs faits rendent très complexes leur implantation algorithmique. Parmi ces difficultés, on compte la résolution d'un système d'équations différentielles et algébriques, la gestion des arcs contraints et non contraints et l'utilisation massive du calcul variationnel pour évaluer les diverses dérivées partielles ( $\mathcal{H}_x, \mathcal{H}_u, \dots$ ). L'ensemble de ces éléments devient difficile à gérer lorsque la dynamique du système est complexe.

Dans le paragraphe suivant, nous allons montrer que ces conditions nécessaires d'optimalité ne sont applicables au cas du guidage que sous certaines hypothèses simplificatrices.

### I.3.1.4 Application au guidage

Pour décrire l'application au guidage des techniques citées dans les précédents paragraphes, nous allons présenter la méthode décrite dans [Li & Bainum 1990] qui permet d'obtenir des trajectoires en temps optimal par itération.

Le critère utilisé est l'intégrale quadratique des couples développés :

$$J = \int_{t_0}^{t_f} \dot{H}_{int}^T R \dot{H}_{int} dt = \int_{t_0}^{t_f} u^T R u dt \quad (1.34)$$

où  $R$  est une matrice de pondération semi-définie positive décomposable sous le forme de Choleski  $R = L^T L \in \mathbb{R}^{m \times m}$ . Considérons  $\lambda_Q$  et  $\lambda_\Omega$  les états adjoints associés à  $Q$  et  $\Omega$  et l'équation d'état donnée par (I.15). Les auteurs proposent alors l'expression de l'Hamiltonien associé aux équations (I.15) et (I.34) suivante :

$$\mathcal{H} = \frac{1}{2} \dot{H}_{int}^T R \dot{H}_{int} + \lambda_Q^T \tilde{\underline{\Omega}} Q + \lambda_\Omega^T (\mathbb{I}_{PF}^{-1} \tilde{\underline{\Omega}} \mathbb{I}_{PF} \Omega + \mathbb{I}_{PF}^{-1} L \dot{H}_{int}) \quad (1.35)$$

où  $\mathbb{I}_{PF}$  est la matrice d'inertie de la plateforme satellite, tandis que  $\tilde{\underline{\Omega}}$  et  $\underline{\underline{\Omega}}$  correspondent aux matrices définies par :

$$\tilde{\underline{\Omega}} = \begin{bmatrix} 0 & r & -q \\ -r & 0 & p \\ q & -p & 0 \end{bmatrix} \quad (1.36)$$

$$\underline{\underline{\Omega}} = \begin{bmatrix} 0 & -\Omega^T \\ \Omega & \tilde{\underline{\Omega}} \end{bmatrix} \quad (1.37)$$

Si  $\lambda_Q$  et  $\lambda_\Omega$  satisfont les conditions nécessaires d'optimalité, il vient :

$$\dot{\lambda}_Q = -\frac{\partial \mathcal{H}}{\partial Q} = \frac{1}{2} \underline{\underline{\Omega}} \lambda_Q \quad (1.38)$$

$$\dot{\lambda}_\Omega = -\frac{\partial \mathcal{H}}{\partial \Omega} = g(\Omega, \lambda_\Omega) + \frac{1}{2} \check{Q} \lambda_Q \quad (1.39)$$

où  $g(\Omega, \lambda_\Omega)$  est un vecteur colonne ( $3 \times 1$ ) défini par :

$$g(\Omega, \lambda_\Omega) = -\frac{\partial (\lambda_\Omega^T \mathbb{I}_{PF}^{-1} \tilde{\underline{\Omega}} \mathbb{I}_{PF} \Omega)}{\partial \Omega} \quad (1.40)$$

et  $\check{Q}$  est la matrice :

$$\check{Q} = \begin{bmatrix} q_1 & q_0 & -q_3 & q_2 \\ q_2 & q_3 & q_0 & -q_1 \\ q_3 & -q_2 & q_1 & -q_0 \end{bmatrix} \quad (1.41)$$

Le principe du maximum de Pontryagin nous donne la condition nécessaire d'optimalité  $\frac{\partial \mathcal{H}}{\partial u} = 0$ , ce qui se traduit par :

$$Ru + L^T \mathbb{I}_{PF}^{-1} \lambda_\Omega = 0 \quad (1.42)$$

en multipliant cette équation par  $(LL^T)^{-1}L$

$$Lu + \mathbb{I}_{PF}^{-1}\lambda_{\Omega} = 0 \quad (1.43)$$

Soit  $L^+$  la pseudo-inverse de  $L$ , une expression de  $u$  répondant aux conditions d'optimalité est alors donnée par :

$$u = -L^+\mathbb{I}_{PF}^{-1}\lambda_{\Omega} = -L^T(LL^T)^{-1}\mathbb{I}_{PF}^{-1}\lambda_{\Omega} \quad (1.44)$$

Il faut maintenant ajouter le problème des contraintes actionneurs qui se traduisent sous forme de contraintes inégalitaires. Les auteurs intègrent la prise en compte des saturations sur les actionneurs dans l'expression de la commande :

$$u_i = \begin{cases} -u_{i,max}\text{sign}(L^{-1}\mathbb{I}_{PF}^{-1}\lambda_{\Omega})_i & \text{si } |(L^{-1}\mathbb{I}_{PF}^{-1}\lambda_{\Omega})_i| \geq u_{i,max} \\ -(L^{-1}\mathbb{I}_{PF}^{-1}\lambda_{\Omega})_i & \text{si } |(L^{-1}\mathbb{I}_{PF}^{-1}\lambda_{\Omega})_i| < u_{i,max} \end{cases}, \quad i = 1, 2, 3 \quad (1.45)$$

A partir de cette formulation l'utilisation d'une méthode de tirs s'avère infructueuse dans le cas général d'arcs saturés non singuliers. Le problème aux deux bouts se résoud en supposant des arcs singuliers, c'est-à-dire quand la matrice d'inertie est diagonale et la matrice  $L$ , correspond à la matrice d'identité. Cette approximation importante permet de simplifier l'équation (1.45) sous la forme suivante :

$$u_i = -\frac{u_{i,max}}{\mathbb{I}_{PF,(i,i)}}\text{sign}(\lambda_{\Omega,i}), \quad i = 1, 2, 3 \quad (1.46)$$

**Remarque I.2** Cette technique induit un certain conservatisme. En effet toutes les raideurs gyroscopiques entraînées par les termes non diagonaux de la matrice d'inertie ne sont pas prises en compte.

Li et Baïnum posent ainsi des conditions d'optimalité appliquées au problème de guidage. Par la suite, ils utilisent des méthodes numériques pour obtenir des trajectoires. Nous présentons dans les paragraphes suivants ces outils numériques.

### I.3.2 Méthodes de tirs

Les méthodes de tirs transforment le problème aux deux bouts initial en un problème de détermination des valeurs initiales. Dans ce type de méthode, la trajectoire du système est intégrée entre l'instant initial et final pour des conditions initiales fixées. La valeur finale de la trajectoire obtenue est alors comparée avec la valeur finale désirée. Le problème consiste alors à déterminer la valeur initiale de la trajectoire qui permettra d'obtenir la valeur finale la plus proche possible de celle désirée. Un algorithme de programmation non linéaire est alors mis en place pour obtenir cette valeur initiale.

#### I.3.2.1 Méthodes indirectes de tirs

Dans le cas d'une méthode indirecte, les entrées de commande sont définies par les conditions d'optimalité (I.28) et les trajectoires par (I.19) et (I.27). Rappelons que l'Hamiltonien est défini par (I.21) ou le cas échéant par (I.32). Le vecteur des variables à optimiser  $\Upsilon$  est formé

par les conditions initiales des états et états adjoints. Pour déterminer ces conditions initiales, on utilise les algorithmes de programmation non linéaire.

La démarche générale pour une méthode indirecte de tirs peut-être résumée comme suit :

*Méthode indirecte de tirs*

*Définir une trajectoire initiale  $t \mapsto (x_0(t), u_0(t), \lambda_0(t))$*   
*tant que les contraintes à l'instant final ne sont pas satisfaites, faire :*

- 1. Discriminer les arcs contraints et non contraints*
- 2. Intégrer les équations (I.19), (I.27), (I.28)*
- 3. Evaluer les contraintes finales*
  - si les contraintes sont respectées, sortir de la boucle*
  - si les contraintes ne sont pas respectées aller en 4*
- 4. Calculer une nouvelle estimation  $x(t_0), \lambda(t_0)$  par des méthodes de gradient (cf. optimisation non linéaire)*

Dans [Li & Bainum 1990], les auteurs ont réussi à mettre en place les conditions nécessaires d'optimalité dérivées du maximum de Pontryagin pour le guidage d'un satellite. Ils obtiennent cependant les solutions présentées sous l'hypothèse d'axisymétrie du véhicule (cf. le paragraphe I.3.1.4). Les auteurs utilisent une méthode indirecte de tirs qui nécessite une trajectoire du système et des co-états,  $t \mapsto (x(t), u(t), \lambda_Q(t), \lambda_\Omega(t))$ , comme solution initiale.

Pour que l'algorithme d'optimisation apporte une solution, la trajectoire initiale doit être suffisamment proche de la solution optimale. Les auteurs définissent cette trajectoire initiale à partir d'une rotation principale après intégration des équations (I.11), (I.12), (I.38), (I.39) et (I.45) et à partir des conditions initiales  $Q(0), \Omega(0), \lambda_Q(0)$  et  $\lambda_\Omega(0)$  dont le calcul est détaillé dans [Li & Bainum 1990]. Enfin ils utilisent un algorithme d'optimisation non linéaire pour obtenir les conditions initiales et ainsi les trajectoires solutions des états adjoints  $\lambda_\Omega$ .

Dans cette approche, la procédure d'optimisation du temps de manoeuvre consiste en une série de cycles. La durée de la manoeuvre est fixée pour chaque cycle. Pour obtenir la durée minimale de manoeuvre, on utilise la règle heuristique suivante : la durée de manoeuvre est considérée comme minimale si toutes les commandes sont saturées durant toute la manoeuvre.

Cependant, cette méthode a été remise en cause par Bilimoria et Wie [Bilimoria & Wie 1993]. Ils ont démontré que le choix de la rotation principale pour trajectoire initiale ne conduit pas à des solutions optimales.

D'autres méthodes basées sur la commande optimale ont été développées puis appliquées au problème de guidage. La section suivante a pour objet de présenter une synthèse de celles-ci.

### **I.3.2.2 Méthodes indirectes de tirs multiples**

Les notions de tirs multiples ont été introduites dans [Bulirsch 1971] pour compenser certains inconvénients des méthodes de tirs. En effet, dans les méthodes de tirs, de petites variations sur les paramètres de début de trajectoire peuvent entraîner des changements importants en fin de

trajectoire, ce qui provoque une grande instabilité numérique dans la procédure d'optimisation. L'espace de temps est divisé en  $N$  intervalles, soit  $N + 1$  noeuds. Les variables d'optimisation sont les valeurs discrétisés des états, des états adjoints et de la commande pour la méthode indirecte. On impose des contraintes supplémentaires de continuité sur les variables d'optimisation en chaque noeud. En posant  $\Upsilon_i$  les valeurs initiales au noeud  $i$  et  $\bar{\Upsilon}_i$  les valeurs finales des trajectoires intégrées sur le segment  $i$ , les contraintes de continuité s'écrivent :

$$c(\Upsilon) = \begin{bmatrix} \Upsilon_2 - \bar{\Upsilon}_1 \\ \vdots \\ \Upsilon_N - \bar{\Upsilon}_{N-1} \end{bmatrix} = 0 \quad (1.47)$$

Cette technique apporte de la robustesse aux méthodes de tirs aussi bien dans le cadre des méthodes indirectes [Bulirsch & Chudej 1991] que directes [Hull & Speyer 1982]. Elle a été reprise avec succès par Bilimoria et Wie [Bilimoria & Wie 1993] dans le cadre de la réorientation d'un satellite dans le cas de la formulation indirecte. Les auteurs utilisent une méthode indirecte de tirs multiples équivalente à celle présentée dans le paragraphe I.3.1.4 dans la mise en place des conditions d'optimalité et proposent une réorientation autour d'un axe pour des angles de  $0$  à  $180^\circ$ . Ainsi, les auteurs étudient différentes structures de commutation dans le cas de commandes saturées afin d'obtenir le temps minimal de manoeuvres. Ils démontrent que la manoeuvre optimale ne se fait pas, de manière générale, autour de l'axe principal de la rotation considérée.

Cependant, des hypothèses fortes sont faites pour obtenir un problème numérique raisonnable : on peut citer l'axisymétrie du véhicule et la contrainte holonome sur les commandes telle que :

$$|u_1|^{2\varepsilon} + |u_2|^{2\varepsilon} + |u_3|^{2\varepsilon} = 1, \quad \varepsilon \geq 1 \quad (1.48)$$

Cette dernière hypothèse réduit le nombre de commandes indépendantes à deux.

Dans le même courant de pensée, on peut citer les travaux de Seywald et Kumar [Seywald & Kumar 1993] qui de façon plus générale ont étudié toutes les commandes possibles pour obtenir des trajectoires en temps minimal, à savoir l'ensemble des arcs singuliers d'ordre fini ou infini. La possibilité d'utiliser uniquement deux entrées de commande pour un satellite d'inertie symétrique a été plus particulièrement mise en avant par Chowdry et Cliff [Chowdry & Cliff 1991]. Ils ont ajouté à ces simplifications, la prise en compte des seules équations de la dynamique (1.12) afin d'obtenir un problème de commande optimale intéressant à résoudre.

Globalement, la difficulté des méthodes de tirs multiples, provient du fait que le nombre de variables augmente proportionnellement avec le nombre de segments temporels. Cela peut entraîner l'accroissement important du temps de résolution du problème par des algorithmes d'optimisation non linéaires.

La principale difficulté de la méthode indirecte de tirs est d'obtenir une trajectoire initiale des états et états adjoints, [Bulirsch, Nerz, Pesch & von Stryk 1993]. En effet, il faut définir une trajectoire initiale suffisamment proche de la condition finale pour obtenir la convergence de la méthode. Or, il est dit dans [Bryson & Ho 1975, page 214] que la valeur finale de la trajectoire est très sensible aux moindres variations des conditions initiales. Les auteurs précisent que les équations de la dynamique étant fortement couplées aux équations d'Euler-Lagrange, il n'est alors pas rare de faire diverger par intégration numériques les trajectoires de l'état et des états adjoints avec de mauvaises conditions initiales.

### I.3.3 Méthodes indirectes de transcription

Les méthodes indirectes de transcription sont semblables aux méthodes indirectes de tirs multiples. Cependant, dans ces dernières, la dynamique est intégrée par un algorithme de résolution d'équations différentielles ordinaires, alors que dans le cas de la transcription, on utilise l'un des schémas implicites d'intégration qui sera présenté dans la section I.4.

Les schémas implicites d'intégration remplacent la contrainte de la dynamique par le calcul d'un vecteur de résidus. La démarche sera exposée en détails dans le paragraphe I.4.1. Ce calcul de résidus porte sur les variables en jeu dans une méthode indirecte à savoir les états, les états adjoints et la commande dont les évolutions sont définies respectivement par (I.19), (I.27), (I.28).

#### *Méthode indirecte de transcription*

*Initialiser  $\Upsilon_i, i = 1, \dots, N + 1$*

*Tant que les contraintes à l'instant final ne sont pas satisfaites faire :*

- 1. Discriminer les arcs contraints et non contraints*
- 2. Calculer les résidus par intégration implicite des équations (I.19), (I.27), (I.28)*
- 3. Evaluer les contraintes finales*
  - si les contraintes sont respectées, sortir de la boucle*
  - si les contraintes ne sont pas respectés aller en 4*
- 4. Calculer une nouvelle estimation  $\tilde{\Upsilon}_i, i = 1, \dots, N + 1$  par des méthodes de gradient (cf. optimisation non linéaire)*

*Calculer le critère de coût*

Les méthodes indirectes de transcription ont été utilisées dans le logiciel COLSYS développé par Ascher et al [Ascher et al. 1979].

Ayant traité de la formulation indirecte du problème de commande optimale, passons maintenant à sa formulation directe

## I.4 Problème de commande optimale : formulation directe

Les approches directes consistent à convertir le problème de commande optimale en un problème d'optimisation paramétrique qui sera résolu par des algorithmes de programmation non linéaire. Le principe est de paramétrer les profils temporels de la commande et/ou de l'état par des courbes paramétrées. Les paramètres de ces courbes forment le vecteur de variables d'optimisation. Les méthodes directes peuvent être classées en deux catégories [Hull 1997] :

**Méthode de tirs** [Kraft 1985, Bock & Plitt 1984], seules les entrées de commande sont considérées comme inconnues et discrétisées. Les trajectoires sont intégrées explicitement.

**Méthodes de transcription** [Hargraves & Paris 1987, Renes 1978], les états et les commandes sont considérés comme inconnus et discrétisés, l'intégration de la dynamique étant

prise en charge par des techniques d'intégration implicite. Il existe aussi un cas implicite où seuls les états sont considérés comme inconnus et discrétisés [Seywald 1994].

Nous étudierons successivement les deux méthodes.

### I.4.1 Méthodes directes de tirs

L'objet de cette méthode est de paramétrer l'évolution des entrées de commande. Il faut donc avoir une expression analytique explicite des commandes telle que, par exemple :

$$u = \rho_1 + \rho_2 t \quad (1.49)$$

ou une relation implicite

$$\rho_1 u(t) + \sin(\rho_2 u(t)) = 0 \quad (1.50)$$

Les paramètres  $\rho_1$  et  $\rho_2$  deviennent alors les variables d'optimisation. On notera  $\Upsilon$  le vecteur des variables d'optimisation. Puisque seules les entrées sont paramétrées, l'état du système est obtenu par intégration de la dynamique du système. De plus, dans une méthode directe de tirs, l'ensemble des contraintes sur la trajectoire et sur les actionneurs est vérifié en un nombre fini de points  $\{\tau_i\}_{i=1,\dots,N}$  :

$$C(\Upsilon) = \begin{pmatrix} c_1(\Upsilon, \tau_1) \\ \vdots \\ c_l(\Upsilon, \tau_N) \\ \vdots \\ c_l(\Upsilon, \tau_1) \\ \vdots \\ c_l(\Upsilon, \tau_N) \end{pmatrix} \geq 0 \quad (1.51)$$

L'algorithme de résolution pour les méthodes de tirs directes peut se résumer comme suit.

*Méthode directe de tirs*

*Initialiser  $\Upsilon$*

*Tant que les contraintes ne sont pas satisfaites, faire :*

1. *Intégrer la trajectoire entre  $t_0$  et  $t_f$*
2. *Evaluer le vecteur de contraintes  $C(\Upsilon)$* 
  - *si les contraintes sont satisfaites : sortir de la boucle*
  - *si les contraintes ne sont pas satisfaites : aller en 3*
3. *Calculer une nouvelle estimée  $\tilde{\Upsilon}$  par des méthodes de gradient (cf. optimisation non linéaire)*

*Calculer le critère de coût*

Les méthodes directes de tirs ont trouvé plusieurs applications dans le domaine spatial. Nous pouvons citer notamment le programme POST (Program to Optimize Simulated Trajectories)

[Brauer, Cornick & Stevenson 1977] développé pour simuler des trajectoires de lanceurs gros porteurs (Titan). Les trajectoires de transfert orbital par impulsion ont pu être aussi traitées par ces méthodes [Betts 1977]. Ce dernier problème a aussi été résolu en considérant une variable de commande continue en temps [Brusch 1979] (manoeuvre orbitale en faible poussée) avec un résultat équivalent.

Toutes les applications qui ont donné des résultats satisfaisants ont un point commun : un faible nombre de variables d'optimisation. En effet, si les entrées de commande  $u$  sont définies par un grand nombre de paramètres, alors les performances liées à la programmation non linéaire s'en trouvent fortement dégradées [Betts 1998]. En effet, les gradients calculés dans les algorithmes d'optimisation non linéaire vont être très sensibles aux variations de ces paramètres. Par exemple, une variation d'un paramètre au début de la trajectoire va se propager jusqu'au bout de la trajectoire. Ce qui a pour effet néfaste de créer des dépendances non linéaires entre les contraintes et les variables d'optimisation.

### I.4.2 Méthodes directes de transcription : collocation

Le comportement des variables d'état étant décrit par des courbes paramétrées, le principe des méthodes directes de transcription, dites de collocation, est de déterminer les paramètres de manière à ce que l'ensemble des contraintes, notamment la dynamique du système, soit vérifié. On note  $\Upsilon$  le vecteur des paramètres.

Le respect de la dynamique du système est assuré par un schéma implicite d'intégration. Celui-ci consiste en l'approximation de la dynamique du système par des courbes paramétrées par  $\Upsilon$  en un nombre d'instant finis appelés points de collocation. Cette approximation se traduit par le calcul d'un résidu  $R$ . Ce résidu est la différence entre l'interpolation de la dynamique  $\dot{x}_i$  et le calcul de la dynamique  $f(x_i, u_i)$  aux points de collocation  $\{\tau_i\}$  :

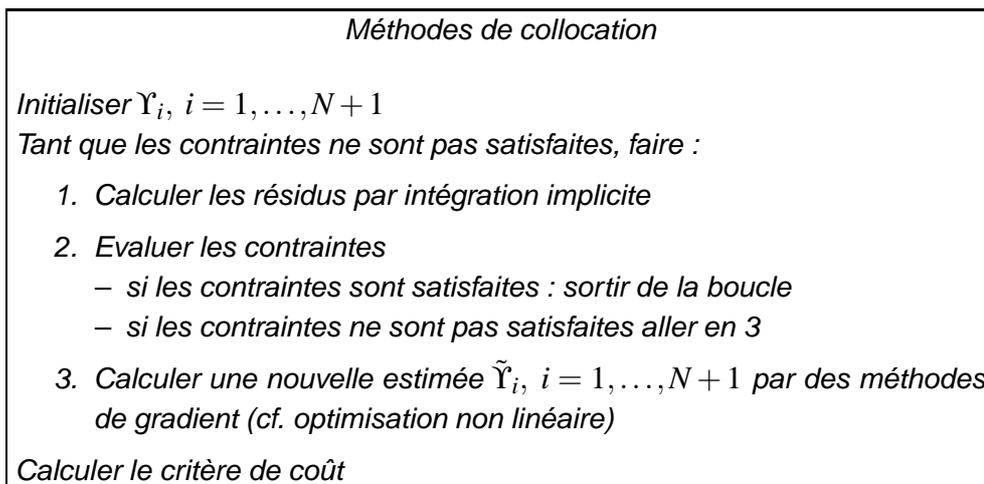
$$R_i = \dot{x}_i - f(x_i, u_i), \quad i = 1, \dots, N_{\tau_i} \quad (1.52)$$

On notera par la suite  $R_i$  le résidu en  $\tau_i$  et  $R$  le vecteur des résidus. Le schéma d'intégration implicite peut être choisi parmi diverses techniques telles que la méthode trapézoïdale, d'Hermite-Simpson [Dickmanns & Well 1975], de Runge-Kutta [Butcher 1964] ou encore de Gauss-Lobatto [Herman & Conway 1996]. Le respect de la dynamique est assuré lorsque l'ensemble des résidus  $R_i$  vérifient un critère d'erreur de type  $\|R\| < \varepsilon$ , ce qui est pris en charge par l'algorithme de programmation non linéaire. Le jeu de contraintes du problème d'optimisation paramétrique est donné par :

$$R(\Upsilon) = 0, \quad C(\Upsilon) \geq 0 \quad (1.53)$$

avec  $C(\Upsilon)$  défini par l'équation (1.51).

Dans le cas où seul l'état est discrétisé, l'approche est similaire à la précédente, le vecteur  $\Upsilon$  ne comportant alors que les valeurs de l'état. Cette approche se pose dans le cadre des systèmes implicites i.e. la dynamique de ces systèmes se définit par l'équation implicite  $F(x, \dot{x}) = 0$ , l'évaluation de l'expression des résidus est alors modifiée. L'algorithme peut se résumer comme suit :



Ces méthodes ont été appliquées à la génération de trajectoires optimales d'attitude par Herman et Conway [Herman & Conway 1992] d'une part et par Scrivener et Thompson dans [Scrivener & Thomson 1993] d'autre part. La différence entre les deux études vient du critère de coût étudié, à savoir la consommation d'énergie pour les premiers et le temps de manoeuvre pour les suivants. Ils basent néanmoins leur méthode de collocation sur la méthode d'Hermite-Simpson i.e. une approximation cubique de l'évolution de l'état entre deux points de segmentation temporelle (cf. figure I.3).

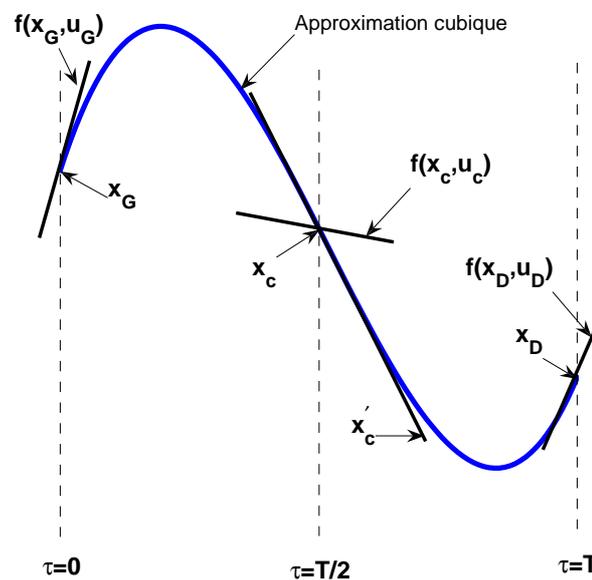


FIG. I.3 – Approximation de la trajectoire d'Hermite-Simpson sur un segment

Dans la technique d'Hermite-Simpson, les points de collocation sont les centres des segments temporels. Ainsi, les coordonnées de l'état aux instants de collocation sont :

$$x_c = \frac{1}{2}(x_G + x_D) + \frac{T}{8}(f_G - f_D) \quad (I.54)$$

où  $T$  est la taille de l'intervalle,  $f_G = f(x_G, u_G)$ ,  $f_D = f(x_D, u_D)$ , les indices  $G$  et  $D$  indiquant respectivement les bornes gauche et droite de l'intervalle considéré. Les entrées de commande varient linéairement sur chaque segment. Ainsi, la commande au centre du segment est donnée par :

$$u_c = \frac{1}{2}(u_G + u_D) \quad (1.55)$$

On calcule par la suite une approximation des équations de la dynamique au centre des segments :

$$f_c = f(x_c, u_c) \quad (1.56)$$

Ensuite, d'après l'approximation cubique de l'évolution des états, on obtient une expression explicite de cette même dynamique :

$$\dot{x}_c = -\frac{3}{2T}(x_G - x_D) - \frac{1}{4}(f_G + f_D) \quad (1.57)$$

Les résidus s'obtiennent alors :

$$R_i = f_{c,i} - \dot{x}_{c,i}, \quad i = 1, \dots, N \quad (1.58)$$

Les autres contraintes sont dûes à la normalisation des quaternions et aux contraintes sur la commande. Le critère de coût est alors :

$$J = \begin{cases} \sum_{i=1}^N \left( \frac{1}{2}(x_{G,i} + x_{D,i})T + TR_i^T R_i \right), & \text{si la consommation de couple est minimisée} \\ (NT - T_f)^2 + T \sum_{i=1}^N R_i^T R_i, & \text{si le temps de manoeuvre est minimisé} \end{cases} \quad (1.59)$$

Le succès des méthodes dites directes est dû principalement à deux faits. Tout d'abord, comme toutes les méthodes directes, elles ne nécessitent pas l'emploi des conditions nécessaires d'optimalité ((1.27), (1.28), (1.29)) et sont donc robustes et faciles à utiliser lorsque l'application devient complexe. En effet, une spécificité de cette formulation est que les conditions d'optimalité sont pris en charge par l'algorithme de programmation non linéaire. Ensuite, contrairement à toutes les autres méthodes présentées, elle ne requiert pas la connaissance d'éventuels arcs de contraintes et de leur séquençement. La discrimination des arcs est supportée par l'algorithme d'optimisation. Ainsi, la discrimination des arcs est éliminée du processus de collocation. En effet, l'algorithme d'optimisation fait implicitement cette discrimination par la stratégie des contraintes actives. Cependant, cela ne permet qu'une approximation des arcs puisqu'eux aussi se définissent sur la grille des points de collocation. Nous n'obtenons alors généralement que des solutions sous-optimales.

Les faiblesses des méthodes de collocation sont inhérentes aux dimensions du problème et à l'algorithme d'optimisation paramétrique non linéaire. En effet, le nombre de variables à optimiser est proportionnel au nombre de segments et au nombre de variables du système. Pour l'exemple précédent on obtient 10 variables (7 variables d'état et 3 entrées de commande).

Les algorithmes d'optimisation nécessitent également de calculer des matrices jacobiennes et hessiennes qui comportent une information sur les dérivées. Ces calculs sont opérés par différences finies. Ces techniques ne sont fiables que si les dimensions du problème sont raisonnables. Si ce n'est pas le cas, il est nécessaire d'utiliser des outils de réduction des gradients [Brenan & Hallman 1994] qui s'avèrent efficaces mais coûteux en temps de calcul.

## I.5 Méthodes composites et autres méthodes

### I.5.1 Association des méthodes de collocation et des méthodes indirectes de tirs multiples

Comme nous l'avons déjà précisé pour les méthodes indirectes, une trajectoire initiale doit être fournie à l'algorithme itératif issu de l'application des conditions nécessaires d'optimalité (I.27), (I.28) et (I.29) et ceci constitue une des difficultés de l'approche. En effet, une solution n'est obtenue que si la trajectoire initiale est suffisamment proche de la solution optimale. Ce qui revient à avoir une connaissance tout au moins partielle de la solution optimale. Le principe introduit par Bulirsch et ses collaborateurs [von Stryk & Bulirsch 1992],[Bulirsch et al. 1993] consiste à calculer cette trajectoire initiale par le biais de méthodes directes de transcription. Ceci présente comme avantage de ne pas avoir besoin d'une solution initiale précise. Selon les auteurs, ces deux méthodes sont complémentaires. En effet, la solution offerte par des méthodes de collocation est obtenue relativement facilement à partir d'une solution initiale peu précise. En revanche, si elle n'est généralement pas optimale, elle en donne une bonne approximation. A partir de cette approximation, les méthodes indirectes prennent le relais et permettent d'aboutir à une trajectoire près de l'optimalité. Shen et Tsiotras [Shen & Tsiotras 1999], à partir des travaux de Chowdry [Chowdhry & Cliff 1991], obtiennent des résultats intéressants en combinant les méthodes directes et indirectes pour un modèle complet.

### I.5.2 Autres méthodes de génération de trajectoires

Nous pouvons évoquer brièvement d'autres techniques comme la méthode de STO (*Switching Time Optimization*) qui a été développée à l'origine dans le cadre de la robotique par Meier et Bryson [Meier & Bryson 1990]. La méthode consiste à optimiser, dans le cadre de commandes saturées, les instants de commutation. Elle a été appliquée au problème de réorientation en temps optimal par Byers [Byers & Vadali 1993] puis par Liu et Singh [Liu & Singh 1996] dans le cadre de trajectoires à consommation minimale. Cette méthode, bien qu'aisément implantable, souffre du fait que le nombre de commutation doit être établi par avance.

## I.6 Conclusion

Nous avons présenté dans ce chapitre une classification des méthodes de commande optimale pour la génération de trajectoires selon deux grands courants : les méthodes indirectes et directes.

Les méthodes indirectes n'apparaissent pas très appropriées pour le problème de guidage. En effet, elles ne sont applicables que sous certaines hypothèses simplificatrices, à savoir l'approximation d'une matrice d'inertie diagonale ou encore la non prise en compte des raideurs gyroscopiques dans le modèle dynamique. De telles hypothèses dans le calcul des profils de commande peuvent entraîner des écarts importants à la trajectoire lors de leur exécution. De plus, les méthodes indirectes nécessitent l'utilisation de variables adjointes, ce qui va engendrer deux types de difficultés. Tout d'abord, les états adjoints augmentent le nombre de variables d'optimisation. Ensuite, il est nécessaire que leur trajectoire initiale soit proche de la trajectoire optimale pour que l'algorithme converge. Or, le calcul de cette trajectoire initiale est déjà un

problème en soi. Les différents contributeurs ont décrit des heuristiques pour pallier ce problème, mais toujours sous un nombre important d'hypothèses simplificatrices.

Les méthodes directes présentent de meilleures dispositions. Elles sont tout d'abord plus simples à mettre en place. Elles ne nécessitent pas d'états adjoints et convergent vers une solution même en cas de trajectoire initiale éloignée de la solution optimale. De plus, dans le cas de la génération de trajectoires d'attitude, elles sont aussi précises que les méthodes indirectes. Cependant, la dimension du problème d'optimisation est grande et peut poser certains problèmes aux algorithmes d'optimisation non linéaires. Ajoutons que les solutions obtenues sont généralement sous optimales mais donnent une bonne approximation de la solution optimale.

Ce tour d'horizon des différents travaux réalisés dans le domaine de la génération de trajectoires par résolution du problème de commande optimale permet de positionner nos travaux et de justifier les orientations suivies par la suite. Dans le chapitre suivant, nous présenterons le concept de la platitude différentielle et son intérêt pour le guidage en attitude des engins spatiaux. Les propriétés de la platitude différentielle permettent de simplifier le problème et rendre très attractives les méthodes directes de commande optimale tout en ne faisant aucune hypothèse sur la dynamique du système, contrairement à l'ensemble des travaux présentés sur l'optimisation de trajectoires d'attitude pour satellites.

---

## Chapitre II

# Platitude différentielle et génération de trajectoires

---

### Sommaire

---

<b>II.1</b>	<b>Introduction</b>	<b>26</b>
<b>II.2</b>	<b>Jets infinis et équivalence de Lie-Bäcklund</b>	<b>26</b>
II.2.1	Jets infinis	27
II.2.1.1	Coordonnées infinies et variétés produits	27
II.2.1.2	Champs de vecteurs	28
II.2.2	Système piloté	29
II.2.3	Équivalence de Lie-Bäcklund	31
<b>II.3</b>	<b>Condition nécessaire et suffisante de platitude différentielle</b>	<b>32</b>
II.3.1	Platitude Différentielle	32
II.3.1.1	Formulation algébrique	32
II.3.1.2	Formulation géométrique	33
II.3.2	Condition nécessaire et suffisante	34
II.3.2.1	Propriétés variationnelles	34
II.3.2.2	Caractérisation algébrique de l'isomorphisme de Lie-Bäcklund	36
II.3.2.3	Intégrabilité	37
II.3.2.4	Théorème de la condition nécessaire et suffisante	38
II.3.2.5	Exemple : véhicule non holonome	39
<b>II.4</b>	<b>Platitude du modèle dynamique d'attitude d'un satellite rigide</b>	<b>40</b>
II.4.1	Modèle dynamique d'attitude d'un satellite rigide	40
II.4.1.1	Cinématique des paramètres d'attitude	40
II.4.1.2	Représentation d'état explicite	43
II.4.2	Applications des conditions nécessaires et suffisantes	44
II.4.3	Sorties plates et bouclage dynamique	45
<b>II.5</b>	<b>Apports de la platitude au problème de commande optimale</b>	<b>46</b>
<b>II.6</b>	<b>Conclusion</b>	<b>47</b>

---

## II.1 Introduction

Ce chapitre est articulé autour de la notion de *platitude différentielle* et ses implications dans les tâches de génération de trajectoires. Ces notions sont importantes pour la compréhension des techniques qui seront mises en oeuvre à partir du chapitre III. En effet, la platitude différentielle constitue une des pierres angulaires de ce mémoire.

Ce chapitre aura pour but, d'une part, de montrer que le modèle dynamique d'attitude d'un satellite rigide possède la propriété de platitude et, d'autre part, de mettre en relief l'ensemble des propriétés utiles pour la construction des algorithmes de génération de trajectoires.

Nous rappellerons les définitions de la platitude différentielle dans deux cadres différents, à savoir celui de la géométrie différentielle [Fliess, Lévine, Martin & Rouchon 1999, van Nieuwstadt, Rathinam & Murray 1998, Martin, Murray & Rouchon 2003] et celui de l'algèbre différentielle [Fliess, Lévine, Martin & Rouchon 1992b, Fliess, Lévine, Martin & Rouchon 1992a, Aranda-Bricaire, Moog & Pomet 1994, Fossas, Franch & Palau 1998]. Les deux approches constituent des cadres d'étude complémentaires. Cependant, nous nous attacherons essentiellement au formalisme de la géométrie différentielle puisqu'il fournit, par le biais de la condition nécessaire et suffisante établie dans [Lévine 2004b], des outils intéressants pour envisager le développement d'un algorithme systématique de démonstration de la platitude à partir d'un modèle dynamique non linéaire. Les notes de cours de Jean Lévine [Lévine 2004a], qui préfigurent un ouvrage en cours d'élaboration, constituent un des documents de référence pour ce chapitre.

La première section du chapitre pose les bases des jets infinis dans le cadre de la géométrie différentielle en dimension infinie et définit la notion d'équivalence de Lie-Bäcklund entre deux systèmes pilotés. Dans un second temps, nous rappellerons la définition de platitude différentielle et la condition nécessaire et suffisante de platitude. Nous détaillerons comment la caractérisation de l'application qui définit l'équivalence de Lie-Bäcklund est au coeur de cette condition nécessaire et suffisante. Ensuite nous appliquerons cette condition au modèle dynamique d'attitude d'un satellite rigide. Enfin, nous reprendrons le problème de génération de trajectoires tel qu'il a été posé dans le chapitre I, à savoir comme un problème de commande optimale. Nous montrerons comment la platitude différentielle transforme ce problème en un problème équivalent, dont nous exploiterons les propriétés dans la suite de ce mémoire.

## II.2 Jets infinis et équivalence de Lie-Bäcklund

Cette première section doit nous amener à appréhender la notion d'équivalence entre deux systèmes comme l'existence d'une application inversible liant toutes les trajectoires d'un système piloté à celles d'un autre système piloté. Les notations de base de géométrie différentielle sont reportées en annexe B afin de ne pas surcharger ce paragraphe.

Fliess et al [Fliess et al. 1999] ont décrit la platitude dans le cadre de la géométrie différentielle en termes d'équivalence de Lie-Bäcklund entre deux systèmes. Du point de vue de la génération de trajectoires, les principales avancées proposées par la platitude sont liées aux propriétés de cette équivalence.

Nous rappellerons la définition des systèmes pilotés non linéaires dans le cadre des jets

infinis sous la forme explicite :

$$\dot{x} = f(x, u) \tag{II.1}$$

où  $x$  appartient à une variété  $X$  de dimension  $n$ ,  $u$  appartient à un ouvert  $U$  de  $\mathbb{R}^m$  et on suppose de plus que le rang de  $\frac{\partial f}{\partial u}$  est constant et égal à  $m$ . Nous considérerons aussi le cas des systèmes non linéaires implicites qui sont décrits sous la forme suivante :

$$F(x, \dot{x}) = 0 \tag{II.2}$$

avec  $x$  appartenant à une variété  $X$  de dimension  $n$  et  $F$  une application  $C^\infty$  de  $TX$  l'espace tangent de cette variété  $X$  dans  $\mathbb{R}^{n-m}$ , dont la matrice jacobienne  $\frac{\partial F}{\partial x}$  est de rang égal à  $n - m$ . Nous préciserons au paragraphe II.2.2 comment il est possible de passer d'une forme à l'autre.

Présentons tout d'abord la topologie des jets infinis.

## II.2.1 Jets infinis

### II.2.1.1 Coordonnées infinies et variétés produits

Les isomorphismes de Lie-Bäcklund sont des applications qui vérifient trois propriétés [Anderson & Ibragimov 1979] :

- chaque composante de ces transformations fait intervenir un nombre fini mais non connu à l'avance des dérivées successives des coordonnées en jeu.
- ces transformations admettent la dérivation par rapport au temps (cf définition II.2).
- ces transformations sont inversibles au sens où, par l'intermédiaire d'une transformation de même type, les coordonnées initiales sont atteignables à partir des coordonnées images.

La première propriété vérifiée par les isomorphismes de Lie-Bäcklund nous amène à considérer un nombre fini mais non prévisible de coordonnées : on parlera ainsi d'une *infinité dénombrable* de coordonnées, que l'on note comme les dérivées successives des coordonnées par rapport au temps :

$$\bar{x} = \left( x_1, \dots, x_n, \dot{x}_1, \dots, \dot{x}_n, \dots, x_1^{(k)}, \dots, x_n^{(k)}, \dots \right) \tag{II.3}$$

$x_i^{(k)}$  désigne la dérivée  $k^{\text{ième}}$  de  $x_i$  par rapport au temps. Ces coordonnées sont naturellement associées à une variété *produit* dite de jets infinis qui est le produit de la variété  $X$  et d'une *infinité dénombrable* de répliques de  $\mathbb{R}^n$  :

$$X \times \mathbb{R}_\infty^n = X \times \mathbb{R}^n \times \mathbb{R}^n \times \dots \tag{II.4}$$

On peut aussi dans le cas explicite où l'entrée  $u$  est spécifiée, introduire les coordonnées :

$$(x, \bar{u}) = \left( x_1, \dots, x_n, u_1, \dots, u_n, \dot{u}_1, \dots, \dot{u}_n, \dots, u_1^{(k)}, \dots, u_n^{(k)}, \dots \right) \tag{II.5}$$

On associe de la même manière ces coordonnées à une variété *produit* de jets infinis :

$$X \times U \times \mathbb{R}_\infty^m = X \times U \times \mathbb{R}^m \times \mathbb{R}^m \times \dots \tag{II.6}$$

Donnons maintenant la définition de la continuité d'une fonction sur les variétés *produits*.

**Définition II.1**

Une fonction  $\phi$  de  $X \times \mathbb{R}_\infty^n$  (resp.  $X \times U \times \mathbb{R}_\infty^m$ ) dans  $\mathbb{R}$  est continue si et seulement si  $\phi$  ne dépend que d'un nombre fini, arbitraire de coordonnées de  $X \times \mathbb{R}_\infty^n$  (resp.  $X \times U \times \mathbb{R}_\infty^m$ ) et si elle est continue par rapport à ses coordonnées.

De plus, pour tout  $k \geq 1$ ,  $\phi \in C^k$  si et seulement si  $\phi$  est  $k$  fois différentiable par rapport à chacune des variables dont elle dépend et si  $\phi^{(k)}$  est continue vis-à-vis de ses coordonnées.

**II.2.1.2 Champs de vecteurs**

Dans le cas d'un système sous la forme implicite, un champ de vecteurs  $C^\infty$  sur  $X \times \mathbb{R}_\infty^n$  est un opérateur différentiel du premier ordre qui peut s'écrire sous la forme suivante :

$$v = \sum_{i \geq 0} \sum_{j=1}^n v_{i,j} \frac{\partial}{\partial x_j^{(i)}} \quad (II.7)$$

Chaque composante  $v_{i,j}$  est une fonction  $C^\infty$  de  $X \times \mathbb{R}_\infty^n$  dans  $\mathbb{R}$ .

De même, dans le cas d'un système sous forme explicite, un champ de vecteurs  $C^\infty$  sur  $X \times U \times \mathbb{R}_\infty^m$  est un opérateur différentiel du premier ordre de la forme

$$w = \sum_{k=1}^n \tilde{w}_k \frac{\partial}{\partial x_k} + \sum_{i \geq 0} \sum_{j=1}^m w_{i,j} \frac{\partial}{\partial u_j^{(i)}} \quad (II.8)$$

dont chaque composante  $\tilde{w}_i, w_{i,j}$  est une fonction  $C^\infty$  de  $X \times U \times \mathbb{R}_\infty^m$  dans  $\mathbb{R}$ . Notons que ces deux définitions sont compatibles avec la notion de dérivée de Lie. Par exemple, soit une fonction  $h$  de classe  $C^\infty$  de  $X \times \mathbb{R}_\infty^n$  dans  $\mathbb{R}$  (resp.  $C^\infty$  de  $X \times U \times \mathbb{R}_\infty^m$  dans  $\mathbb{R}$ ), sa dérivée de Lie dans la direction  $v$  (resp.  $w$ ) est donnée par

$$L_v h = \sum_{i \geq 0} \sum_{j=1}^n v_{i,j} \frac{\partial h}{\partial x_j^{(i)}}, \quad \text{dans le cadre d'une fonction implicite}$$

$$L_w h = \sum_{k=1}^n \tilde{w}_k \frac{\partial h}{\partial x_k} + \sum_{i \geq 0} \sum_{j=1}^m w_{i,j} \frac{\partial h}{\partial u_j^{(i)}} \quad \text{dans le cadre d'une fonction explicite}$$

Parmi les champs de vecteurs de premier ordre, nous pouvons distinguer les champs de vecteurs dits de Cartan qui nous permettront de définir les systèmes pilotés.

**Définition II.2 (champ de Cartan)**

**Formulation implicite** Le champ de Cartan dit trivial est défini sur  $X$  par

$$\tau_X = \sum_{i \geq 0} \sum_{j=1}^n x_j^{(i+1)} \frac{\partial}{\partial x_j^{(i)}} \quad (II.9)$$

Le champ trivial  $\tau_X$  correspond au système différentiel trivial pour tout  $j$ , dont n'importe quelle fonction de classe  $C^\infty$  du temps  $t \mapsto x(t)$  sur  $X$  est solution. De plus, la dérivée de Lie d'une fonction arbitraire  $h$  est donnée par :

$$L_{\tau_X} h = \sum_{i \geq 0} \sum_{j=1}^n x_j^{(i+1)} \frac{\partial h}{\partial x_j^{(i)}} = \frac{dh}{dt} \quad (II.10)$$

On peut alors associer à  $\tau_X$  l'opérateur de dérivation  $\frac{d}{dt}$ .

**Formulation explicite** Dans le cas des systèmes explicites, il nous faut utiliser la variété infinie  $X \times U \times \mathbb{R}_\infty^m$  pour définir le champ de vecteurs de Cartan  $\bar{f}$  défini par

$$\bar{f} = \sum_{k=1}^n f_k \frac{\partial}{\partial x_k} + \sum_{i \geq 0} \sum_{j=1}^m u_i^{(j+1)} \frac{\partial}{\partial u_j^{(i)}} \quad (\text{II.11})$$

où  $f_k$  est la  $k^{\text{me}}$  composante de l'application  $f$ .

Cette présentation des jets infinis étant faite, elle va nous permettre de passer à la définition d'un système piloté dans le cadre de la géométrie différentielle en dimension infinie.

### II.2.2 Système piloté

Nous pouvons définir simplement un système piloté dans sa formulation dite interne :

**Définition II.3 (système piloté : formulation interne)**

Un système piloté est donnée par une paire  $(\mathfrak{X}, \bar{f})$  où  $\mathfrak{X} = X \times U \times \mathbb{R}_\infty^m$  et  $\bar{f}$  est le champ de Cartan

En d'autres termes, un système explicite piloté est donné par

$$\begin{cases} \dot{x} &= f(x, u) \\ \dot{u} &= \dot{u} \\ \ddot{u} &= \ddot{u} \\ &\vdots \end{cases} \quad (\text{II.12})$$

Donnons maintenant la définition d'un système piloté dans sa formulation externe i.e. pour un système implicite :

**Définition II.4 (système piloté : formulation externe)**

Un système implicite est donnée par le triplet  $(\mathfrak{X}, \tau_X, F)$ , où la variété  $\mathfrak{X} = X \times \mathbb{R}_\infty^n$  est muni du champ de Cartan trivial  $\tau_X$  défini par (II.9) et  $F$  est défini par (II.2).

Les deux formulations des systèmes pilotés sont équivalentes. On peut exprimer un système implicite (II.2) défini par le champ de Cartan (II.9) dans les coordonnées de la formulation externe (II.3) comme un système explicite sur les coordonnées de la formulation interne. D'après le théorème des fonctions implicites dont on pourra trouver un rappel en annexe, il existe une application  $\tilde{f}$  de classe  $C^\infty$  d'un ouvert de  $X \times \mathbb{R}^m$  vers  $\mathbb{R}^m$  telle que

$$\begin{aligned} \dot{x}_{m+1} &= \tilde{f}_1(x, \dot{x}_1, \dots, \dot{x}_m) \\ &\vdots \\ \dot{x}_n &= \tilde{f}_{n-m}(x, \dot{x}_1, \dots, \dot{x}_m) \end{aligned}$$

Remarquons que dans l'expression précédente les variables  $\dot{x}_1, \dots, \dot{x}_m$  sont libres. On peut alors poser  $u_1 = \dot{x}_1, \dots, u_m = \dot{x}_m$ , il vient ainsi :

$$f(x, u) = \begin{pmatrix} u_1 \\ \vdots \\ u_m \\ \tilde{f}_1(x, u) \\ \vdots \\ \tilde{f}_{n-m}(x, u) \end{pmatrix} \quad (\text{II.13})$$

On retrouve la formulation interne (II.12) en remarquant que pour tout  $j \geq 0$  on a  $u_i^{(j)} = x_i^{(j+1)}$  avec  $i = 1, \dots, m$ . On retrouve de même les coordonnées définies par (II.5) en remplaçant  $\dot{x}, \ddot{x}, \dots$  par  $f(x, u), \frac{d}{dt}(f(x, u)), \dots$

A l'inverse, un système explicite défini par le champ de vecteurs (II.11) dans les coordonnées (II.5) peut s'écrire comme un système implicite dans les coordonnées externes (II.3). Tout d'abord, nous savons que le rang de  $\frac{\partial f}{\partial u}$  est égal à  $m$ . Par permutation des lignes nous plaçons aux  $m$  premières, les  $m$  lignes indépendantes de la jacobienne. Cette nouvelle matrice jacobienne correspond à  $\frac{\partial \tilde{f}}{\partial u}$  avec  $\tilde{f}$  tel que :

$$\tilde{f}_i = f_{\bar{\omega}(i)}$$

avec  $i = 1, \dots, n$  et  $\bar{\omega}$  l'application bijective de  $1, \dots, n$  vers  $1, \dots, n$  qui définit les permutations effectuées précédemment. On notera le vecteur permuté d'état  $\tilde{x}$  défini par  $\tilde{x}_i = x_{\bar{\omega}(i)}$  avec  $i = 1, \dots, n$ . Puisque les  $m$  premières lignes de  $\frac{\partial \tilde{f}}{\partial u}$  sont indépendantes, le théorème des fonctions implicites nous permet d'inverser le système d'équations

$$\begin{cases} \dot{\tilde{x}}_1 = \tilde{f}_1(\tilde{x}, u) \\ \vdots \\ \dot{\tilde{x}}_m = \tilde{f}_m(\tilde{x}, u) \end{cases} \quad (\text{II.14})$$

afin d'exprimer la commande  $u$  :

$$u = U(\tilde{x}, \dot{\tilde{x}}_1, \dots, \dot{\tilde{x}}_m) \quad (\text{II.15})$$

En réinjectant l'expression précédente de  $u$  dans celle de  $\tilde{f}(\tilde{x}, u)$  et en éliminant les  $m$  premières équations qui sont alors identiquement vérifiées, les  $n - m$  équations restantes sont de la forme :

$$F_i(\tilde{x}, \dot{\tilde{x}}) = \dot{\tilde{x}}_i - \tilde{f}_i(\tilde{x}, U(\tilde{x}, \dot{\tilde{x}}_1, \dots, \dot{\tilde{x}}_m)) = 0, \quad i = m + 1, \dots, n \quad (\text{II.16})$$

avec le rang de  $\frac{\partial F}{\partial \dot{\tilde{x}}}$  égal à  $n - m$ . Ainsi le champ de vecteurs (II.16) exprimé dans les coordonnées (II.3) est un système implicite.

Cette définition d'un système piloté nous permet maintenant de décrire l'équivalence de Lie-Bäcklund.

### II.2.3 Equivalence de Lie-Bäcklund

Soient deux systèmes pilotés dans leur formulation interne  $(\mathfrak{M}, \bar{f})$  et  $(\mathfrak{N}, \bar{g})$  où  $\mathfrak{M} = X \times U \times \mathbb{R}_\infty^m$ ,  $X$  désigne une variété de dimension  $n$  et  $U$  une variété de dimension  $m$ ,  $\mathfrak{N} = Y \times V \times \mathbb{R}_\infty^q$ ,  $Y$ , correspondant à une variété de dimension  $p$  et  $V$  une variété de dimension  $q$ ,  $\bar{f}$  et  $\bar{g}$  sont les champs de Cartan associés.

#### Définition II.5 (Equivalence de Lie-Bäcklund : formulation interne [Fliess et al. 1999])

Soient les points  $(x_0, \bar{u}_0) \in \mathfrak{M}$  et  $(y_0, \bar{v}_0) \in \mathfrak{N}$ . Les deux systèmes  $(\mathfrak{M}, \bar{f})$  et  $(\mathfrak{N}, \bar{g})$  sont localement équivalents au sens de Lie-Bäcklund s'il existe :

- une application de classe  $C^\infty$  au voisinage de  $(y_0, \bar{v}_0)$   $\Phi : \mathfrak{N} \mapsto \mathfrak{M}$  telle que  $(x_0, \bar{u}_0) = \Phi(y_0, \bar{v}_0)$  et vérifiant que l'image du champ de Cartan  $\bar{g}$  par  $\Phi$  notée  $\Phi_* \bar{g}$  corresponde au champ  $\bar{f}$
- et inversement une application inverse de classe  $C^\infty$  au voisinage de  $(x_0, \bar{u}_0)$   $\Psi : \mathfrak{M} \mapsto \mathfrak{N}$  telle que  $(y_0, \bar{v}_0) = \Psi(x_0, \bar{u}_0)$  et vérifiant que l'image du champ de Cartan  $\bar{f}$  par  $\Psi$  notée  $\Psi_* \bar{f}$  corresponde au champ  $\bar{g}$

Les applications,  $\Phi$  et  $\Psi$ , sont appelées isomorphismes réciproques locaux de Lie-Bäcklund.

#### Définition II.6 (Equivalence de Lie-Bäcklund : formulation externe [Lévine 2004b])

Considérons les systèmes implicites  $(\mathfrak{X}, \tau_X, F)$ ,  $\mathfrak{X} = X \times \mathbb{R}_\infty^n$ , dans un voisinage  $V$  de  $\bar{x}_0 \in \mathfrak{X}_0$  défini par

$$\mathfrak{X}_0 = \left\{ \bar{x} \in X \times \mathbb{R}_\infty^n \mid L_{\tau_X}^k (F(x, \dot{x})) = 0, \forall k \geq 0 \right\} \quad (\text{II.17})$$

où  $\frac{\partial F}{\partial \dot{x}}$  est de rang plein (i.e. toutes les lignes sont indépendantes entre elles) dans  $V$ , et  $(\mathfrak{Y}, \tau_Y, G)$ ,  $\mathfrak{X} = Y \times \mathbb{R}_\infty^p$ , dans un voisinage  $W$  de  $\bar{y}_0 \in \mathfrak{Y}_0$  défini par

$$\mathfrak{Y}_0 = \left\{ \bar{x} \in Y \times \mathbb{R}_\infty^p \mid L_{\tau_Y}^k (G(z, \dot{z})) = 0, \forall k \geq 0 \right\} \quad (\text{II.18})$$

où  $\frac{\partial G}{\partial \dot{z}}$  est de rang plein (i.e. toutes les lignes sont indépendantes entre elles) dans  $W$ . Les deux systèmes implicites sont équivalents au sens de Lie-Bäcklund s'il existe :

- une application de classe  $C^\infty$  et inversible de  $W$  dans  $V$   $\Phi : W \mapsto V$  telle que  $\bar{x}_0 = \Phi(\bar{y}_0)$  et  $\Phi_* \tau_Y = \tau_X$
- son application inverse de classe  $C^\infty$  et inversible de  $V$  dans  $W$   $\Psi : V \mapsto W$  telle que  $\bar{y}_0 = \Psi(\bar{x}_0)$  et  $\Psi_* \tau_X = \tau_Y$

On appelle image réciproque, ou induite, d'une forme  $\omega$  définie sur la variété  $X$  par l'application  $\varphi$  la forme  $\nu$  défini par  $\varphi_* \omega$ . Les applications  $\Phi$  et  $\Psi$  sont appelées isomorphismes réciproques de Lie-Bäcklund.

Le théorème suivant donne une propriété intéressante de l'équivalence de Lie-Bäcklund qui permettra de justifier par la suite la dimension des sorties plates.

**Théorème II.1** Si deux systèmes sont Lie-Bäcklund équivalents alors ils ont le même nombre d'entrées indépendantes ■

La preuve de ce théorème est disponible dans [Fliess et al. 1999].

**Remarque II.1** Deux systèmes Lie-Bäcklund équivalents ne sont pas nécessairement de même dimension d'état.

Rappelons maintenant une propriété associée à l'équivalence de Lie-Bäcklund qui aura un impact sur le problème de génération de trajectoires (voir paragraphe II.5).

**Théorème II.2** *L'équivalence de Lie-Bäcklund préserve les points d'équilibre* ■

### Démonstration

Supposons que les systèmes  $\dot{x} = f(x, u)$  et  $\dot{z} = g(z, v)$  sont Lie-Bäcklund équivalents. Soit  $(x_0, u_0)$  un point d'équilibre i.e.  $f(x_0, u_0) = 0, u_0^{(j)} = 0, \forall j \geq 1$ . Comme

$$\begin{aligned} y_0 &= \Psi_0(x_0, u_0, \dot{u}_0, \dots, u_0^{(s)}) = \Psi_0(x_0, u_0, 0, \dots, 0) \\ v_0^{(j)} &= \Psi_{j+1}(x_0, u_0, \dot{u}_0, \dots, u_0^{(s+j+1)}) = \Psi_{j+1}(x_0, u_0, 0, \dots, 0), \forall j \end{aligned}$$

il vient alors

$$\begin{aligned} \dot{z}_0 &= \frac{\partial \Psi_0}{\partial x} \dot{x}_0 + \frac{\partial \Psi_0}{\partial u} \dot{u}_0 + \dots + \frac{\partial \Psi_0}{\partial u^{(s)}} u_0^{(s+1)} = 0 \\ v_0^{(j+1)} &= \frac{\partial \Psi_{j+1}}{\partial x} \dot{x}_0 + \frac{\partial \Psi_{j+1}}{\partial u} \dot{u}_0 + \dots + \frac{\partial \Psi_{j+1}}{\partial u^{(s+j+1)}} u_0^{(s+j+2)} = 0, \forall j \end{aligned}$$

Nous venons ainsi de prouver que  $(y_0, v_0)$  est un point d'équilibre de  $\dot{z} = g(z, v)$ . Un procédé identique permet de montrer que l'image d'un point d'équilibre de  $\dot{y} = g(y, v)$  est un point d'équilibre de  $\dot{x} = f(x, u)$ . □

Voyons maintenant comment l'équivalence de Lie-Bäcklund est au coeur de la condition nécessaire et suffisante de la platitude différentielle [Lévine 2004b]).

## II.3 Condition nécessaire et suffisante de platitude différentielle

Dans un premier temps, nous allons définir la platitude différentielle dans le cadre de l'algèbre différentielle puis dans le cadre des variétés des jets infinis et de l'équivalence de Lie-Bäcklund. Dans la seconde définition, le corollaire 1 prendra toute son importance dans le problème de commande optimale défini par la suite (cf. paragraphe II.5). Puis nous exposerons la condition nécessaire et suffisante de platitude qui a été formalisée par Lévine [Lévine 2004b]. Le niveau de détail doit permettre au lecteur intéressé d'appliquer cette condition à tout système non linéaire. Ainsi, un exemple académique sera présenté afin de résumer la démarche à suivre.

### II.3.1 Platitude Différentielle

#### II.3.1.1 Formulation algébrique

La platitude différentielle a été introduite pour la première fois par Fliess et al. en 1992 [Fliess et al. 1992b, Fliess et al. 1992a]. Les auteurs utilisent le formalisme de l'algèbre différentielle dans la mesure où un système  $y$  est vu comme une extension différentielle d'un nombre fini de commandes indépendantes. Alors un système est dit plat s'il existe un jeu de variables, appelées sorties plates, tel que le système soit algébrique par rapport à l'extension différentielle générée

par les sorties plates. En d'autres termes, un système est plat s'il existe un jeu de  $m$  sorties plates  $z$  et leurs dérivées successives qui déterminent les  $n$  états et les  $m$  entrées du système sans intégration tel que :

$$x = \Psi_x(z, \dot{z}, \ddot{z}, \dots, z^{(s)}) \quad (\text{II.19})$$

$$u = \Psi_u(z, \dot{z}, \ddot{z}, \dots, z^{(s+1)}) \quad (\text{II.20})$$

mais tel qu'on ait aussi la réciproque :

$$z = h(x, u, \dot{u}, \ddot{u}, \dots, u^{(r)}) \quad (\text{II.21})$$

Cette définition a été décrite pour la première fois dans [Fliess et al. 1992b]. Cette définition de la platitude différentielle nous permettra d'exprimer les contraintes sur l'état et l'entrée de commande en termes de sortie plate (voir paragraphe II.4.3).

### II.3.1.2 Formulation géométrique

La définition de la platitude différentielle présentée ici a d'abord été établie dans [Fliess et al. 1999] pour les systèmes explicites puis a été étendue aux systèmes implicites dans [Lévine 2004b]. La condition nécessaire et suffisante de platitude développée dans la partie II.3.2 découle de la définition de la platitude pour les systèmes implicites.

#### Définition II.7

Le système explicite  $(\mathfrak{M}, F)$  (resp. le système implicite  $(\mathfrak{X}, \tau_x, F)$ ) est plat en  $(x_0, \bar{u}_0) \in \mathfrak{M}$  (resp.  $x_0, \in \mathfrak{X}$ ) si et seulement s'il est Lie-Bäcklund équivalent en ce point à un système trivial explicite  $(\mathbb{R}_\infty^m, \tau_m)$  (resp. un système trivial implicite  $(\mathbb{R}_\infty^m, \tau_m, 0)$ ) où  $\tau_m$  est le champ de vecteurs trivial sur  $\mathbb{R}_\infty^m$  muni des coordonnées

$$\left( z_1, \dots, z_m, \dot{z}_1, \dots, \dot{z}_m, \dots, z_1^{(k)}, \dots, z_m^{(k)}, \dots \right)$$

tel que

$$\tau_m = \sum_{i \geq 0} \sum_{j=1}^m x_j^{(i+1)} \frac{\partial}{\partial x_j^{(i)}} \quad (\text{II.22})$$

Le vecteur  $z = (z_1, \dots, z_m)$  est appelé sortie plate.

Notons que le théorème II.1 permet de justifier que le nombre de composantes d'une sortie plate est égal au nombre d'entrées indépendantes du système plat.

**Corollaire 1** Les composantes de la sortie plate sont donc les états du système trivial. D'après la définition d'un système trivial II.2, les  $m$  composantes de  $z$  sont différentiellement indépendantes : il n'y a pas de relation identiquement vérifiée entre  $z_i$  et  $z_j, \dot{z}_j, \ddot{z}_j, \dots, z_j^{(s+1)}$ ,  $i \neq j$ . Nous en déduisons deux propriétés fortes pour le problème de génération de trajectoires :

1. toute fonction de classe  $C^\infty$  vérifie la dynamique de la sortie plate,
2. un espace des sorties plates,  $O_{\bar{z}}$ , peut-être défini avec les coordonnées  $\{z, \dot{z}, \ddot{z}, \dots, z^{(s)}\}$ . ■

### II.3.2 Condition nécessaire et suffisante

Considérons un système implicite  $(\mathcal{X}, \tau_X, F)$  et  $\mathcal{X}_0$  un ouvert de  $\mathcal{X}$ . Nous admettrons que ce système est régulier au sens où  $\text{rang} \left( \frac{\partial F}{\partial \bar{x}} \right) = n - m$  avec  $n$  la dimension de la variété  $X$  qui compose la variété de jets infinis  $\mathcal{X} = X \times \mathbb{R}_\infty^n$ .

Dans le paragraphe II.3.2.1 la définition de la platitude II.7 sera reformulée en termes de différentielle exacte de l'application  $F$ , ce qui amènera à une définition de la platitude dans le formalisme d'une algèbre matricielle mais non commutative. Les conditions nécessaires et suffisantes de Lévine [Lévine 2004b] s'articulent autour de la caractérisation de l'isomorphisme de Lie-Bäcklund  $\Phi$  d'une part, et de l'intégrabilité d'un idéal particulier d'autre part. Le paragraphe II.3.2.2 exploitera la définition donnée dans le paragraphe précédent pour caractériser l'isomorphisme de Lie-Bäcklund. Dans le paragraphe II.3.2.3, le problème de l'intégrabilité est posé avant de conclure sur le théorème qui établit les conditions nécessaires et suffisantes dans le paragraphe II.3.2.4.

#### II.3.2.1 Propriétés variationnelles

Le système implicite  $(\mathcal{X}, \tau_X, F)$  est plat s'il vérifie la définition II.7. Il existe alors des isomorphismes inverses de Lie-Bäcklund  $\Phi$  et  $\Psi$ . Nous allons tenter de les décrire en fonction de la différentielle de  $F$ .

La forme différentielle exacte des composantes de  $F$  s'écrit :

$$dF_i = \sum_{j=1}^n \left( \frac{\partial F_i}{\partial x_j} dx_j + \frac{\partial F_i}{\partial \dot{x}_i} d\dot{x}_i \right), \quad i = 1, \dots, n - m \quad (\text{II.23})$$

Notons que ces formes différentielles s'expriment dans les bases des espaces tangents et cotangents de la variété  $\mathcal{X}$  qui sont respectivement :

$$\left\{ \begin{array}{l} \frac{\partial}{\partial x_i^{(j)}} | i = 1, \dots, n, j \geq 0 \\ dx_i^{(j)} | i = 1, \dots, n, j \geq 0 \end{array} \right\}$$

On appelle 1-forme toute combinaison linéaire des bases de l'espace cotangent à savoir  $\sum_i v_i d\omega_i$  où  $v_i$  sont des fonctions méromorphes et  $\omega_i$ , les composantes de  $\bar{x}$ . Le théorème suivant permet de lier l'isomorphisme de Lie-Bäcklund  $\Phi$  à la forme différentielle exacte  $dF$  de  $F$ . Une démonstration de ce théorème est disponible dans la référence [Lévine 2004b].

**Théorème II.3** *Un système implicite  $(\mathcal{X}, \tau_X, F)$  est plat en  $(\bar{x}_0, \bar{z}_0) \in \mathcal{X} \times \mathbb{R}_\infty^m$  si et seulement si il existe  $\Phi$  un difféomorphisme inversible et continu, dont l'inverse est aussi continu, de  $\mathbb{R}_\infty^m$  vers  $\mathcal{X}$  satisfaisant  $\Phi(\bar{z}_0) = \bar{x}_0$  et tel que*

$$\Phi^* dF = 0 \quad (\text{II.24})$$

où l'opérateur  $*$  correspond à l'image réciproque d'une 1-forme de l'espace cotangent par une application. ■

Détaillons la condition (II.24), nous avons :

$$\begin{aligned}
 \Phi^* dF &= \sum_{i \geq 0} \sum_{j=1}^m \left( \frac{\partial F}{\partial x} \frac{\partial \phi_0}{\partial z_j^{(i)}} + \frac{\partial F}{\partial \dot{x}} \frac{\partial \phi_1}{\partial z_j^{(i)}} \right) dz_j^{(i)} \\
 &= \sum_{i \geq 0} \sum_{j=1}^m \left( \frac{\partial F}{\partial x} \frac{\partial \phi_0}{\partial z_j^{(i)}} dz_j^{(i)} + \frac{\partial F}{\partial \dot{x}} \frac{d}{dt} \left( \frac{\partial \phi_0}{\partial z_j^{(i)}} dz_j^{(i)} \right) \right) \\
 &= \left( \frac{\partial F}{\partial x} + \frac{\partial F}{\partial \dot{x}} \frac{d}{dt} \right) \Big|_{\bar{x}=\Phi(\bar{z})} \left( \sum_{i \geq 0} \sum_{j=1}^m \frac{\partial \phi_0}{\partial z_j^{(i)}} \frac{d^i}{dt^i} dz_j \right) \quad (II.25)
 \end{aligned}$$

Posons les matrices polynomiales où la variable est l'opérateur différentiel  $\frac{d}{dt}$  :

$$P(F) = \frac{\partial F}{\partial x} + \frac{\partial F}{\partial \dot{x}} \frac{d}{dt} \quad (II.26)$$

$$P(\phi_0) = \sum_{i \geq 0} \frac{\partial \phi_0}{\partial z^{(i)}} \frac{d^i}{dt^i} \quad (II.27)$$

avec  $P(F)$  est de dimension  $(n-m) \times n$  et  $P(\phi_0)$  de dimension  $n \times m$ . L'équation (II.24) s'écrit alors :

$$\Phi^* dF|_{\bar{z}} = P(F)|_{\Phi(\bar{z})} P(\phi_0)|_{\bar{z}} dz = 0 \quad (II.28)$$

ou encore

$$\Phi^* dF|_{\Psi(\bar{x})} = P(F)|_{\bar{x}} P(\phi_0)|_{\Psi(\bar{x})} dz = 0 \quad (II.29)$$

Par construction, les singletons de ces matrices sont des polynômes en  $\frac{d}{dt}$  dont les coefficients sont des fonctions méromorphes de  $\mathfrak{X}$  dans  $\mathbb{R}$ . Une fonction méromorphe sur  $\mathbb{R}$  est continue et bornée sur  $\mathbb{R}$  sauf peut-être en un nombre fini de points.

Avec  $\mathfrak{K}$  l'anneau des fonctions méromorphes<sup>1</sup> de  $\mathfrak{X}$  dans  $\mathbb{R}$  (cf. annexe B), on note  $\mathfrak{K} \left[ \frac{d}{dt} \right]$  l'anneau principal des polynômes en  $\frac{d}{dt}$  à coefficients dans  $\mathfrak{K}$ . De même, avec  $\mathfrak{K}_{\mathfrak{N}}$  le champ des fonctions méromorphes de  $\mathfrak{N}$  dans  $\mathbb{R}$  et  $\mathfrak{N} = \mathbb{R}_{\infty}^m$ , on note alors  $\mathfrak{K}_{\mathfrak{N}} \left[ \frac{d}{dt} \right]$  l'anneau principal des polynômes en  $\frac{d}{dt}$  à coefficients dans  $\mathfrak{K}_{\mathfrak{N}}$ . Il est important pour la suite de remarquer que les anneaux  $\mathfrak{K} \left[ \frac{d}{dt} \right]$  et  $\mathfrak{K}_{\mathfrak{N}} \left[ \frac{d}{dt} \right]$  sont des algèbres non commutatifs. En effet l'opérateur  $\frac{d}{dt}$  est non commutatif :

$$\frac{d}{dt} x - x \frac{d}{dt} = \dot{x}$$

L'ensemble de ces conditions implique que la fonction  $F$  est elle-même méromorphe. Nous supposons que les applications  $\Phi$  et  $\Psi$  sont également méromorphes.

On définit  $\mathcal{M}_{p,q} \left[ \frac{d}{dt} \right]$  comme étant l'ensemble des matrices de dimension  $p \times q$  sur  $\mathfrak{K} \left[ \frac{d}{dt} \right]$ .  $\mathcal{M}_{p,q} \left[ \frac{d}{dt} \right]$  constitue un module sur  $\mathfrak{K} \left[ \frac{d}{dt} \right]$ .

Dans le cas général, l'inverse d'une matrice carré inversible de  $\mathcal{M}_{p,p} \left[ \frac{d}{dt} \right]$  n'appartient pas forcément à  $\mathcal{M}_{p,p} \left[ \frac{d}{dt} \right]$ . Les matrices  $\mathcal{M}_{p,p} \left[ \frac{d}{dt} \right]$  dont l'inverse appartient à  $\mathcal{M}_{p,p} \left[ \frac{d}{dt} \right]$  sont qualifiées d'unimodulaires. Cet ensemble est noté  $\mathcal{U}_p \left[ \frac{d}{dt} \right]$  tel que  $\mathcal{U}_p \left[ \frac{d}{dt} \right] \subset \mathcal{M}_{p,p} \left[ \frac{d}{dt} \right]$ .

Les éléments de  $\mathcal{M}_{p,q} \left[ \frac{d}{dt} \right]$  admettent une réduction diagonale encore appelée décomposition de Smith [Cohn 1985].

<sup>1</sup>Une fonction défini sur un ensemble  $D$  est méromorphe si elle y est définie sur cet ensemble à l'exception éventuelle de certains points

**Théorème II.4 (décomposition de Smith)** Soit une matrice polynomiale  $A \in \mathcal{M}_{p,q} \left[ \frac{d}{dt} \right]$ , il existe deux matrices  $V \in \mathcal{U}_p \left[ \frac{d}{dt} \right]$  et  $U \in \mathcal{U}_q \left[ \frac{d}{dt} \right]$  telles que

$$VAU = \begin{cases} \begin{pmatrix} \Delta_p & 0_{p,q-p} \end{pmatrix} & \text{si } p < q, \\ \Delta_p & \text{si } p = q, \\ \begin{pmatrix} \Delta_q \\ 0_{p-q,q} \end{pmatrix} & \text{si } p > q, \end{cases} \quad (\text{II.30})$$

Les matrices  $V \in \mathcal{U}_{n-m} \left[ \frac{d}{dt} \right]$  et  $U \in \mathcal{U}_n \left[ \frac{d}{dt} \right]$  ne sont pas uniques mais  $\Delta_{n-m} \in \mathcal{M}_{n-m,n-m} \left[ \frac{d}{dt} \right]$  est une matrice diagonale unique telle que les éléments  $d_{i,i}$  divisent  $d_{j,j}$  pour tout  $i, j$  tel que  $1 \leq i \leq j \leq n-m$ . ■

### Définition II.8

On dit qu'une matrice  $M \in \mathcal{M}_{p,q} \left[ \frac{d}{dt} \right]$  est hyper-régulière si et seulement si sa décomposition de Smith donne  $\Delta = I$ .

Dans le cas traité ici, nous avons  $P(F) \in \mathcal{M}_{n-m,n} \left[ \frac{d}{dt} \right]$  : elle admet donc une décomposition de Smith telle que

$$VP(F)U = (\Delta_{n-m}, 0_{n-m,m}) \quad (\text{II.31})$$

### II.3.2.2 Caractérisation algébrique de l'isomorphisme de Lie-Bäcklund

Il s'agit de mettre en évidence l'existence d'une solution à l'équation matricielle  $P(F)\Theta dz = 0$  avec  $\Theta \in \mathcal{M}_{n,m} \left[ \frac{d}{dt} \right]$ . Supposons maintenant que  $P(F)$  soit hyper-régulière i.e. elle vérifie

$$VP(F)U = (I_m, 0_{n-m,m}) \quad (\text{II.32})$$

On notera que  $U \in D - \text{Smith}(P(F))$  et  $V \in G - \text{Smith}(P(F))$  du fait de la non unicité des matrices  $U$  et  $V$ . Dans ce cadre, l'ensemble de toutes les matrices  $P(\phi_0)|_{\Psi(\bar{x}) \in \mathcal{M}_{n,m} \left[ \frac{d}{dt} \right]}$  admissibles vis-à-vis des contraintes (II.28) et (II.29) peut être complètement caractérisé par la résolution de l'équation

$$P(F)\Theta dz = 0 \quad (\text{II.33})$$

dans laquelle les éléments de  $\Theta \in \mathcal{M}_{n,m} \left[ \frac{d}{dt} \right]$  sont supposés ne pas être liés à  $\phi_0$ .

**Lemme 1** Posons l'ensemble des matrices hyper-régulières  $\Theta \in \mathcal{M}_{n,m} \left[ \frac{d}{dt} \right]$  satisfaisant (II.33). Cet ensemble n'est pas vide et se définit par

$$\Theta = U \begin{pmatrix} 0_{n-m,m} \\ I_m \end{pmatrix} W \quad (\text{II.34})$$

avec  $U \in D - \text{Smith}(P(F))$  et  $W \in \mathcal{U}_m \left[ \frac{d}{dt} \right]$  choisie arbitrairement. ■

### Démonstration

On peut écrire l'équation (II.33) grâce à (II.32) telle que :

$$VP(F)UU^{-1}\Theta = (I_m, 0_{n-m,m})U^{-1}\Theta = 0 \quad (\text{II.35})$$

On en déduit que  $U^{-1}\Theta = \begin{pmatrix} 0_{n-m,m} \\ I_m \end{pmatrix} W$  où  $W \in \mathcal{U}_m \left[ \frac{d}{dt} \right]$  est choisie arbitrairement ou encore  $\Theta = U \begin{pmatrix} 0_{n-m,m} \\ I_m \end{pmatrix} W$ . L'hyper-régularité de  $\Theta$  et donc son inversibilité implique l'hyper-régularité de  $W$ .  $\square$

En notant  $\hat{U} = U \begin{pmatrix} 0_{n-m,m} \\ I_m \end{pmatrix}$ , il vient alors  $\Theta = \hat{U}W$ .

**Lemme 2** Pour toute matrice  $Q \in G - \text{Smith}(\hat{U})$ , il existe  $Z \in \mathcal{U}_m \left[ \frac{d}{dt} \right]$  telle que

$$Q\Theta = \begin{pmatrix} I_m \\ 0_{n-m,m} \end{pmatrix} Z \quad (\text{II.36})$$

A fortiori, il existe une matrice  $\hat{Q} = (0_{n-m,m}, I_{n-m})Q$  telle que  $\hat{Q}\Theta = 0$  équivalente à  $P(F)$  (i.e.  $\exists L \in \mathcal{U}_{n-m} \left[ \frac{d}{dt} \right]$  telle que  $P(F) = L\hat{Q}$ ).  $\blacksquare$

### Démonstration

Nous savons que la matrice  $\hat{U}$  est hyper-régulière du fait de l'hyper-régularité de  $U$  et de l'équation  $\hat{U} = U \begin{pmatrix} 0_{n-m,m} \\ I_m \end{pmatrix}$ . En appliquant la décomposition de Smith à la matrice  $\hat{U}$ , nous avons

une matrice  $Q \in \mathcal{U}_n \left[ \frac{d}{dt} \right]$  et  $R \in \mathcal{U}_m \left[ \frac{d}{dt} \right]$  telles que  $Q\hat{U}R = \begin{pmatrix} I_m \\ 0_{n-m,m} \end{pmatrix}$ .

La première partie du résultat s'obtient en posant  $R^{-1}W = Z \in \mathcal{U}_m \left[ \frac{d}{dt} \right]$ . Ensuite, en multipliant l'équation à gauche (II.36) par  $(0_{n-m,m}, I_{n-m})$ , il vient :

$$(0_{n-m,m}, I_{n-m})Q\Theta = (0_{n-m,m}, I_{n-m}) \begin{pmatrix} I_m \\ 0_{n-m,m} \end{pmatrix} Z \quad (\text{II.37})$$

En posant  $Q = \begin{pmatrix} \tilde{Q} \\ \hat{Q} \end{pmatrix}$ , il vient  $\hat{Q} = (0_{n-m,m}, I_{n-m})Q$ , on a alors :

$$\hat{Q}\Theta = 0 \quad (\text{II.38})$$

En comparant les équations (II.33) et (II.38) et en tenant compte de l'hyper-régularité de  $\hat{U}$ , on déduit l'existence d'une matrice unimodulaire  $L$  telle que  $P(F) = L\hat{Q}$   $\square$

### II.3.2.3 Intégrabilité

La notion qui sera développée dans ce paragraphe peut être rapprochée de la notion d'accessibilité d'un système (i.e. le principe de commandabilité pour un système non linéaire). Le second point des conditions nécessaires et suffisantes va consister à montrer que le système implicite obtenu à travers l'isomorphisme de Lie-Bäcklund est intégrable et que l'ensemble des trajectoires est fonction des coordonnées du premier système implicite.

En notant  $Q_{i,j}$  le polynôme en  $\frac{d}{dt}$  de l'entrée  $(i, j)$  de  $Q$ , nous pouvons décrire le vecteur  $\omega$  de  $m$  1-formes défini à partir de  $Q$  tel que

$$\omega(\bar{x}) = \begin{pmatrix} \omega_1(\bar{x}) \\ \vdots \\ \omega_m(\bar{x}) \end{pmatrix} = \begin{pmatrix} \sum_{j=1}^n \sum_{k \geq 0} Q_{1,j}^k(\bar{x}) dx_j^{(k)} \\ \vdots \\ \sum_{j=1}^n \sum_{k \geq 0} Q_{m,j}^k(\bar{x}) dx_j^{(k)} \end{pmatrix} = \tilde{Q} dx \quad (\text{II.39})$$

L'indépendance des éléments de  $\omega(\bar{x})$  est la conséquence de l'hyper-régularité de  $\hat{Q}$ . Nous pouvons alors générer un idéal sur l'anneau  $\mathcal{R}\left[\frac{d}{dt}\right]$  que nous noterons  $\Xi$ . Rappelons que le  $\mathcal{R}\left[\frac{d}{dt}\right]$ -idéal  $\Xi$  est l'ensemble de toutes les combinaisons à coefficients dans  $\mathcal{R}\left[\frac{d}{dt}\right]$  des formes  $\eta \wedge \omega_i$ ,  $\eta$  étant une  $p$ -forme arbitraire avec  $i = 1, \dots, m$ .

### Définition II.9

Le  $\mathcal{R}\left[\frac{d}{dt}\right]$ -idéal  $\Xi$  généré par les 1-formes  $\omega_1, \dots, \omega_m$  est fortement fermé si et seulement si il existe une matrice  $M \in \mathcal{U}_m\left[\frac{d}{dt}\right]$  telle que  $d(M\omega) = 0$ .

#### II.3.2.4 Théorème de la condition nécessaire et suffisante

Dans ce cadre, nous pouvons donner la condition nécessaire et suffisante de platitude.

**Théorème II.5** Une condition nécessaire et suffisante pour qu'un système implicite  $(\mathcal{X}, \tau_{\mathcal{X}}, F)$  soit plat en  $(\bar{x}_0, \bar{z}_0)$  est qu'il existe deux matrices  $U \in D - \text{Smith}(P(F))$  et  $Q \in G - \text{Smith}(\hat{U})$  avec  $\hat{U} = U \begin{pmatrix} 0_{n-m, m} \\ I_m \end{pmatrix}$  telles que le  $k\left[\frac{d}{dt}\right]$ -idéal  $\Xi$  généré par les 1-formes  $\omega_1, \dots, \omega_m$  définies par (II.39) est fortement clos au voisinage de  $\bar{x}_0$  dans  $\mathcal{X}$ . ■

Une preuve de ce théorème est disponible dans [Lévine 2004b]. Les idées directrices de la démonstration sont les suivantes.

**Condition nécessaire :** en se basant sur le théorème II.3, nous pouvons, par la décomposition de Smith de  $P(\phi_0)$  et le jeu des transformations décrites dans les lemmes 1 et 2, écrire :

$$Z^{-1}\hat{Q}dx = dz \quad (\text{II.40})$$

où  $dx$  et  $dz$  sont les différentiels des coordonnées du système implicite  $x$  et du système trivial  $z$ . En notant  $M = Z^{-1}$  et en définissant le vecteur  $\omega$  tel que (II.39), nous pouvons écrire que  $M\omega = dz$ . Enfin en prenant la différentielle extérieure de chaque coté de l'égalité nous obtenons la condition de forte fermeture du  $\mathcal{R}\left[\frac{d}{dt}\right]$ -idéal  $\Xi$  i.e.  $d(M\omega) = 0$ .

**Condition suffisante :** on considère pour acquis les conditions du théorème à savoir l'existence des matrices  $U \in D - \text{Smith}(P(F))$  et  $Q \in G - \text{Smith}(\hat{U})$  et tel que  $\Omega$  soit fortement fermé. La condition de forte fermeture permet de définir un vecteur de 1-formes  $\eta$  défini par  $\eta = M\omega$  tel que  $\eta$  est une 1-forme exacte par le lemme de Poincaré. On peut ainsi définir une application  $\psi_0$  telle que  $d\psi_0 = \eta$ . En notant  $z = \psi_0(\bar{x})$ ,  $\bar{x} \in \mathcal{X}_0$ , il reste à démontrer que l'application  $\Psi = \left(\psi_0, \frac{d}{dt}\psi_0, \frac{d^2}{dt^2}\psi_0 \dots\right) = (\psi_0, \psi_1, \psi_2, \dots)$  est un isomorphisme de Lie-Bäcklund tel que  $\bar{z} = \Psi(\bar{x})$ .

Le grand intérêt de cette condition nécessaire et suffisante est qu'elle caractérise la platitude d'un système implicite à partir d'une analyse matricielle de la différentielle exacte de celui-ci. Cette analyse fait appel à des outils qui sont formalisés d'un point de vue algorithmique. Ces conditions nécessaires et suffisantes mettent aussi en avant la possibilité de définir algébriquement l'isomorphisme de Lie-Bäcklund qui lie les états du système aux sorties plates. Cependant, la grande difficulté provient de la non commutativité des anneaux sur l'opérateur  $\frac{d}{dt}$ . Ainsi, les enjeux et les difficultés de cette méthode font de ce problème un champ de recherche important au sein de l'automatique non linéaire. Des travaux sur la génération automatisée de sorties plates dans le cas linéaire font l'objet d'un projet INRIA [Quadrat 2007]

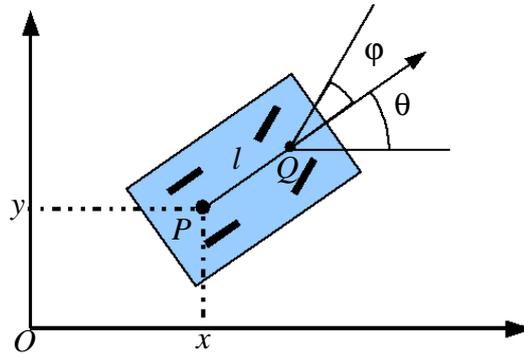


FIG. II.1 – Véhicule non holonome roulant sans glisser

### II.3.2.5 Exemple : véhicule non holonome

Considérons un véhicule à 4 roues roulant sans glissement dans le plan  $(O, \vec{x}, \vec{y})$ . Les coordonnées du point  $P$ , milieu de l'essieu arrière sont notées  $(x, y)$ . Soit  $Q$  le point milieu de l'essieu avant, on note  $l$  la distance entre  $P$  et  $Q$ ,  $\theta$  l'angle entre l'axe du véhicule et l'axe  $(O, \vec{x})$  et  $\varphi$  l'angle de braquage des roues. Les conditions de roulement sans glissement s'expriment par le fait que  $\frac{d\vec{O}P}{dt}$  est parallèle à  $\vec{P}Q$  d'une part et d'autre part que  $\frac{d\vec{O}Q}{dt}$  est parallèle aux roues avant. En notant  $u = \left\| \frac{d\vec{O}P}{dt} \right\| \cdot \frac{\vec{P}Q}{\|\vec{P}Q\|}$ , la cinématique du véhicule est donnée par :

$$\begin{aligned} \dot{x} &= u \cos \theta \\ \dot{y} &= u \sin \theta \\ \dot{\theta} &= \frac{u}{l} \tan \varphi \end{aligned} \quad (\text{II.41})$$

Mis sous forme implicite, ce système devient :

$$F = \dot{x} \sin \theta - \dot{y} \cos \theta = 0 \quad (\text{II.42})$$

Sa différentielle est donnée par :

$$dF\left(\frac{d}{dt}\right) = \left( \sin \theta \frac{d}{dt} \quad -\cos \theta \frac{d}{dt} \quad \dot{x} \cos \theta + \dot{y} \sin \theta \right) \quad (\text{II.43})$$

On a alors

$$dF\left(\frac{d}{dt}\right) \begin{pmatrix} dx \\ dy \\ d\theta \end{pmatrix} \quad (\text{II.44})$$

La décomposition de Smith de  $dF$  consiste à définir une matrice  $U$  et  $V$  telles que :

$$V dF U = (\Delta \quad 0 \quad 0) \quad (\text{II.45})$$

avec  $\Delta$  un scalaire. En prenant,  $V = 1$  et  $U$  telle que

$$U = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & \frac{\cos \theta}{\dot{x} \cos \theta + \dot{y} \sin \theta} \frac{d}{dt} & -\frac{\sin \theta}{\dot{x} \cos \theta + \dot{y} \sin \theta} \frac{d}{dt} \end{pmatrix} \quad (\text{II.46})$$

il vient que  $\delta = \dot{x} \cos \theta + \dot{y} \sin \theta$  qui est bien un scalaire vis-à-vis de la variable  $\frac{d}{dt}$ . L'expression de  $\hat{U}$  est alors donnée par :

$$\hat{U} = U \begin{pmatrix} 0_{1,2} \\ I_2 \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ 1 & 0 \\ \frac{\cos \theta}{\Delta} \frac{d}{dt} & -\frac{\sin \theta}{\Delta} \frac{d}{dt} \end{pmatrix} \quad (\text{II.47})$$

La décomposition de Smith de  $\hat{U}$  permet d'obtenir la matrice  $Q \in G - \text{Smith}(\hat{U})$  :

$$Q = \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ \frac{\sin \theta}{\Delta} \frac{d}{dt} & -\frac{\cos \theta}{\Delta} \frac{d}{dt} & 1 \end{pmatrix} \quad (\text{II.48})$$

Ainsi le vecteur des 1-formes  $\omega$  se définit par :

$$\begin{pmatrix} \omega_1 \\ \omega_2 \end{pmatrix} = \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \end{pmatrix} \begin{pmatrix} dx \\ dy \\ d\theta \end{pmatrix} \quad (\text{II.49})$$

La condition de forte fermeture de la définition II.9 est trivialement vérifiée en prenant la matrice  $M$  égale à la matrice identité de dimension 2. L'idéal  $\Xi$  engendré par  $(dx, dy)$  est donc fortement fermé.

Notons enfin que les propriétés de platitude de ce système ont été utilisées dans le cadre de la synthèse de lois de commande robuste [d'Andrea Novel, Campion & Bastin 1995, Villagra, d'Andrea Novel, Mounier & Pengov 2005].

Appliquons maintenant la condition nécessaire et suffisante à un modèle dynamique d'attitude d'un satellite rigide.

## II.4 Platitude du modèle dynamique d'attitude d'un satellite rigide

Nous allons démontrer dans cette section que le modèle dynamique d'attitude d'un satellite rigide est différentiellement plat.

Commençons tout d'abord par choisir nos paramètres d'attitude afin de donner le modèle d'état du satellite rigide.

### II.4.1 Modèle dynamique d'attitude d'un satellite rigide

#### II.4.1.1 Cinématique des paramètres d'attitude

Les équations de la cinématique sont obtenues par l'écriture de l'équation de propagation des paramètres d'attitude. Ainsi le modèle satellite dépend de la représentation d'attitude que nous choisirons. Comme nous l'avons vu dans la section I.2 du chapitre précédent, la représentation la plus commune est celle par quaternion normalisé. La propagation des quaternions

d'attitude est décrite par (I.5).

$$\begin{bmatrix} \dot{q}_0 \\ \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \end{bmatrix} = \frac{1}{2} \begin{bmatrix} q_0 & -q_1 & -q_2 & -q_3 \\ q_1 & q_0 & -q_3 & q_2 \\ q_2 & q_3 & q_0 & -q_1 \\ q_3 & -q_2 & q_1 & -q_0 \end{bmatrix} \begin{bmatrix} 0 \\ p \\ q \\ r \end{bmatrix} \quad (\text{II.50})$$

Rappelons que les paramètres  $q_i$  sont les composantes du quaternion d'attitude  $Q$  et  $p, q, r$  sont les trois composantes du vecteur vitesse de rotation  $\Omega$ . Bien que l'utilisation de l'équation (I.5) soit largement répandue pour décrire la cinématique, leur utilisation pose un problème du point de vue de la platitude. En effet, l'inversion de modèle (cf paragraphe II.4.3) qui consiste à exprimer les variables d'état en fonction des sorties plates, nécessite d'inverser l'équation de la cinématique (I.5). Or, le déterminant de la matrice de propagation,  $q_0$ , est nul pour une rotation de  $\pi \pmod{2\pi}$  radians.

Nous allons maintenant proposer une famille de représentation d'attitude qui répond à la contrainte précédemment citée. Cette famille est dérivée de la représentation par quaternion normé. En effet, ces paramètres d'attitude sont obtenus par projection stéréographique de la sphère de dimension 4 sur laquelle évolue un quaternion normé sur un hyperplan de dimension 3. Il en résulte un jeu de 3 paramètres stéréographiques, dits paramètres de Rodrigues, possédant un point de singularité dans leur représentation de la sphère des quaternions et par conséquent de l'attitude.

Les paramètres de Rodrigues dépendent du point et du plan de projection, il en est de même pour le point de singularité [Schaub & Junkins 1996]. Il est donc possible de choisir la projection qui permettra de rejeter au plus loin la singularité. Dans ce contexte, le choix se porte sur les paramètres stéréographiques symétriques dit Paramètres Modifiés de Rodrigues (PMR) notés  $\sigma_i$  [Tsiotras, Shen & Hall 2001]. En effet les PMR trouvent leur point de singularité en  $\pm 2\pi$ . Autrement dit, les PMR sont capables de décrire toute rotation à l'exception des révolutions complètes.

Dans le cas où la révolution complète devait être représentée au cours d'une mission, il est possible d'utiliser un jeu de paramètres duaux  $\sigma_i^S$ . Ces deux jeux, bien que distincts représentent la même attitude. Les PMR duaux  $\sigma^S$  trouvent leur point de singularité pour la rotation nulle. Il est ainsi possible de décrire toute rotation sans rencontrer de singularité par le biais d'une méthode adéquate de passage entre les paramètres duaux.

De plus, les PMR ont un certain nombre d'avantages sur le plan numérique. Les non linéarités qu'ils présentent sont de type polynomiales dans la description de leur propagation et de la matrice de rotation. La relation entre les quaternions normalisés et les PMR décrit la projection stéréographique des quaternions sur l'hyperplan médian  $q_0 = 0$  (cf figure II.2). Elle se résume de la façon suivante :

$$\sigma_i = \frac{q_i}{1 + q_0} \quad i = 1, 2, 3 \quad (\text{II.51})$$

$$q_i = \frac{2\sigma_i}{1 + \sigma^T \sigma} \quad i = 1, 2, 3 \quad (\text{II.52})$$

$$q_0 = \frac{1 - \sigma^T \sigma}{1 + \sigma^T \sigma} \quad i = 1, 2, 3 \quad (\text{II.53})$$

$$(\text{II.54})$$

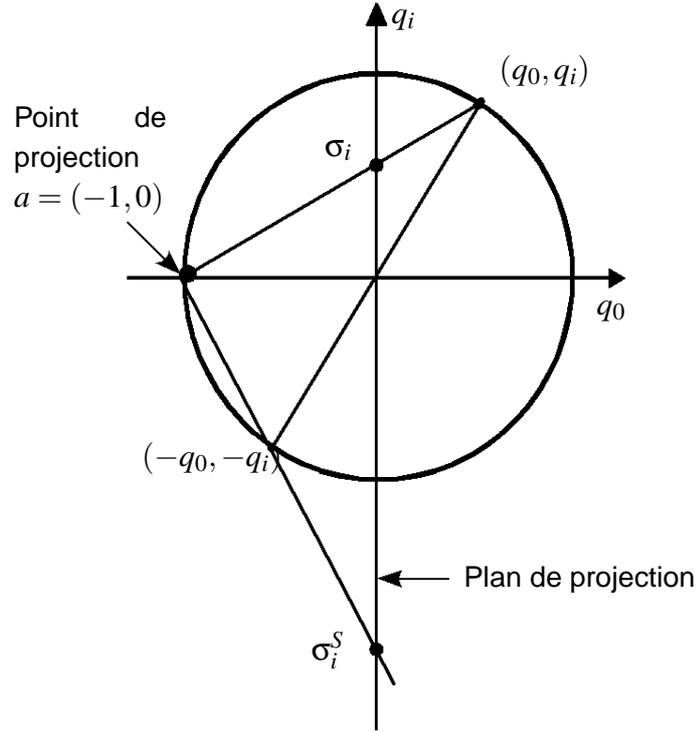


FIG. II.2 – Le paramètre modifié de Rodrigues  $\sigma_i$  et son dual sont des projections stéréographiques du quaternion d'attitude et de son opposé.

Les PMR s'expriment aussi en fonction de la rotation principale  $\mathcal{R}$  définie par l'angle  $\theta$  et le vecteur unitaire  $e = [e_1 e_2 e_3]^T$

$$\sigma = e \tan\left(\frac{\theta}{4}\right) \quad (II.55)$$

L'équation de propagation des PMR est donnée par

$$\dot{\sigma} = \frac{1}{2} \left[ I \left( \frac{1 - \sigma^T \sigma}{2} \right) + [\tilde{\sigma}] + \sigma^T \sigma \right] \Omega \quad (II.56)$$

où  $I$  correspond à la matrice identité,  $\sigma$  désigne le vecteur des PMR et  $[\tilde{\sigma}]$  représente l'opérateur matriciel suivant :

$$[\tilde{\sigma}] = \begin{bmatrix} 0 & -\sigma_3 & \sigma_2 \\ \sigma_3 & 0 & -\sigma_1 \\ -\sigma_2 & \sigma_1 & 0 \end{bmatrix} \quad (II.57)$$

Ainsi, l'équation (II.56) de la cinématique peut s'écrire sous forme matricielle :

$$\begin{aligned} \dot{\sigma} &= \frac{1}{4} \begin{bmatrix} 1 + \sigma_1^2 - \sigma_2^2 - \sigma_3^2 & 2(\sigma_1 \sigma_2 - \sigma_3) & 2(\sigma_1 \sigma_3 + \sigma_2) \\ 2(\sigma_1 \sigma_2 + \sigma_3) & 1 - \sigma_1^2 + \sigma_2^2 - \sigma_3^2 & 2(\sigma_2 \sigma_3 - \sigma_1) \\ 2(\sigma_1 \sigma_3 - \sigma_2) & 2(\sigma_2 \sigma_3 + \sigma_1) & 1 - \sigma_1^2 - \sigma_2^2 + \sigma_3^2 \end{bmatrix} \Omega \\ &= \Theta(\sigma) \Omega \end{aligned} \quad (II.58)$$

Avec cette paramétrisation, le déterminant de la matrice  $\Theta(\sigma)$  est un polynôme pair qui n'a pas de valeur nulle, ce qui constitue le principal avantage de cette paramétrisation. La matrice de

rotation en termes de PMR s'écrit :

$$C(\sigma) = \frac{1}{(1 + \sigma^T \sigma)^2} \begin{bmatrix} 4(\sigma_1^2 - \sigma_2^2 - \sigma_3^2) + \Sigma^2 & 8\sigma_1\sigma_2 + 4\sigma_3\Sigma & 8\sigma_1\sigma_3 - 4\sigma_2\Sigma \\ 8\sigma_1\sigma_2 - 4\sigma_3\Sigma & 4(-\sigma_1^2 + \sigma_2^2 - \sigma_3^2) + \Sigma^2 & 8\sigma_2\sigma_3 + 4\sigma_1\Sigma \\ 8\sigma_1\sigma_3 + 4\sigma_2\Sigma & 8\sigma_2\sigma_3 - 4\sigma_1\Sigma & 4(-\sigma_1^2 - \sigma_2^2 + \sigma_3^2) + \Sigma^2 \end{bmatrix} \quad (\text{II.59})$$

où  $\Sigma = 1 - \sigma^T \sigma$ .

Nous définissons les PMR duaux par rapport aux PMR originaux de la manière suivante :

$$\sigma_i^S = \frac{-\sigma_i}{\sigma^T \sigma} \quad i = 1, 2, 3 \quad (\text{II.60})$$

Remarquons que les PMR et leurs duaux donnent exactement la même matrice de rotation.

On peut définir de la même façon la propagation des PMR duaux. En notant  $\sigma^2 = \sigma^T \sigma$ , la dynamique des PMR duaux s'écrit :

$$\dot{\sigma}^S = \frac{1}{\sigma^4} \left( \dot{\sigma} - \frac{1}{2}(1 + \sigma^2) [\tilde{\sigma}] \Omega - (1 - \sigma^4) \frac{\Omega}{4} \right) \quad (\text{II.61})$$

**Remarque II.2** Une méthode de passage entre les PMR et ses duaux peut être déduite de l'équation (II.60). En effet, on peut écrire à partir de cette équation :

$$\|\sigma\|_2 = \frac{1}{\|\sigma^S\|_2} \quad (\text{II.62})$$

Ainsi, il paraît intéressant de commuter entre les deux représentations au moment où leurs normes sont égales, ce qui permettra de conserver les paramètres utilisés au sein de la sphère unité de dimension 3.

#### II.4.1.2 Représentation d'état explicite

Tout d'abord, définissons les états que nous prendrons en compte. S'il paraît évident que les paramètres d'attitude et les composantes du vecteur vitesse de rotation sont des variables d'état, le moment cinétique interne sera considéré comme une entrée. En effet, au regard de l'équation (I.10), le couple moteur n'est autre que la dérivée du moment cinétique. Nous obtenons les états et les entrées de commandes suivantes :

$$x = [\sigma_1, \sigma_2, \sigma_3, p, q, r]^T \quad (\text{II.63})$$

$$u = [\dot{H}_{int,1}, \dot{H}_{int,2}, \dot{H}_{int,3}]^T \quad (\text{II.64})$$

Ainsi pour ce modèle,  $n = 6$  et  $m = 3$ .

Nous allons maintenant rappeler l'équation  $\dot{x} = f(x, u)$  qui définit le modèle explicite du satellite à partir des équations (II.58), (I.10) :

$$\begin{bmatrix} \dot{\sigma}_1 \\ \dot{\sigma}_2 \\ \dot{\sigma}_3 \\ \dot{\Omega} \end{bmatrix} = \begin{bmatrix} (1 + \sigma_1^2 - \sigma_2^2 - \sigma_3^2)p + 2(\sigma_1\sigma_2 - \sigma_3)q + 2(\sigma_1\sigma_3 + \sigma_2)r \\ 2(\sigma_1\sigma_2 + \sigma_3)p + (1 - \sigma_1^2 + \sigma_2^2 - \sigma_3^2)q + 2(\sigma_2\sigma_3 - \sigma_1)r \\ 2(\sigma_1\sigma_3 - \sigma_2)p + 2(\sigma_2\sigma_3 + \sigma_1)q + (1 - \sigma_1^2 - \sigma_2^2 + \sigma_3^2)r \\ -\mathbb{I}_{PF}^{-1}(\dot{H}_{int} + \Omega \wedge H_{tot}) \end{bmatrix} \quad (\text{II.65})$$

Appliquons alors la condition nécessaire et suffisante de platitude au modèle dynamique d'attitude d'un satellite rigide.

## II.4.2 Applications des conditions nécessaires et suffisantes

**Lemme 3** *Le modèle dynamique d'un satellite rigide décrit par (II.65) est différentiellement plat et admet le vecteur  $(\sigma_1, \sigma_2, \sigma_3)$  comme sortie plate.* ■

### Démonstration

Afin de démontrer ce résultat, nous allons appliquer la condition nécessaire et suffisante du théorème II.5. Il nous faut décrire le système sous forme implicite, le système explicite étant donné par (II.65), nous avons :

$$\dot{\sigma} = \Theta(\sigma)\Omega \quad (\text{II.66})$$

$$-\mathbb{I}_{PF}\dot{\Omega} = \dot{H}_{int} + \Omega \wedge H_{tot} \quad (\text{II.67})$$

Afin d'obtenir le modèle implicite équivalent au modèle (II.66), (II.67) nous reprenons la méthode utilisée pour montrer l'équivalence des formulations interne et externe (cf le paragraphe II.2.2). Nous obtenons ainsi la forme implicite de notre modèle qui correspond à l'équation (II.66) :

$$\begin{aligned} F(x, \dot{x}) &= \dot{\sigma} - \Theta(\sigma)\Omega = 0 \\ &= \begin{bmatrix} \dot{\sigma}_1 - \frac{1}{4} \left( (1 + \sigma_1^2 - \sigma_2^2 - \sigma_3^2)p - 2(\sigma_1\sigma_2 + \sigma_3)q - 2(\sigma_1\sigma_3 - \frac{1}{2}\sigma_2)r \right) \\ \dot{\sigma}_2 - \frac{1}{4} \left( 2(\sigma_1\sigma_2 - \sigma_3)p - (1 - \sigma_1^2 + \sigma_2^2 - \sigma_3^2)q - 2(\sigma_3\sigma_2 + \sigma_1)r \right) \\ \dot{\sigma}_3 - \frac{1}{4} \left( 2(\sigma_1\sigma_3 + \sigma_2)p - 2(\sigma_3\sigma_2 - \sigma_1)q - (1 - \sigma_1^2 - \sigma_2^2 + \sigma_3^2)r \right) \end{bmatrix} \end{aligned} \quad (\text{II.68})$$

Ce résultat se justifie de différentes manières. Avec ce choix de  $F$  nous vérifions que  $\text{rang} \left( \frac{\partial F}{\partial \dot{x}} \right) = n - m$ . Ainsi d'après le théorème des fonctions implicites, nous avons  $n - m$  composantes dérivées qui s'expriment en fonction des  $m$  autres que l'on peut considérer comme libres. On retrouve ici un résultat assez classique qui permet aux algorithmes de guidage rencontrés dans l'industrie de n'utiliser que des profils de vitesses pour déterminer les trajectoires d'attitude sans se soucier du profil de couples associés. Calculons maintenant la forme différentielle exacte de  $F$  :

$$\begin{aligned} dF &= \frac{\partial F}{\partial x} dx + \frac{\partial F}{\partial \dot{x}} d\dot{x} \\ &= \frac{1}{4} \begin{bmatrix} -2\sigma_1 * p - 2\sigma_2 q - 2\sigma_3 r + 4 \frac{d}{dt} & 2\sigma_2 p - 2\sigma_1 q + 2r \\ -2\sigma_2 p + 2\sigma_1 q - 2r & -2\sigma_1 p - 2\sigma_2 q - 2\sigma_3 r + \frac{d}{dt} \\ -2\sigma_3 p + 2q + 2\sigma_1 r & -2p - 2\sigma_3 q + 2\sigma_2 r \\ 2\sigma_3 p - 2q - 2\sigma_1 r & -1 - \sigma_1^2 + \sigma_2^2 + \sigma_3^2 \\ \dots & 2p + 2\sigma_3 q - 2\sigma_2 r & -2\sigma_1 \sigma_2 + 2\sigma_3 \\ -2\sigma_1 p - 2\sigma_2 q - 2\sigma_3 r + \frac{d}{dt} & -2\sigma_1 \sigma_3 - 2\sigma_2 \\ -2\sigma_1 \sigma_2 - 2\sigma_3 & -2\sigma_1 \sigma_3 + 2\sigma_2 \\ \dots & -1 + \sigma_1^2 - \sigma_2^2 + \sigma_3^2 & -2\sigma_3 \sigma_2 - 2\sigma_1 \\ -2\sigma_3 \sigma_2 + 2\sigma_1 & -1 + \sigma_1^2 + \sigma_2^2 - \sigma_3^2 \end{bmatrix} dx \\ &= P(F)dx \end{aligned} \quad (\text{II.69})$$

En décomposant  $P(F)$ , nous obtenons grâce à des outils de calcul formel [Morio 2007] :

$$VP(F)U = (I_m, 0_{n-m,m}) \quad (\text{II.70})$$

$P(F)$  est donc bien hyper-régulière. Ainsi, d'après les lemmes 1 et 2, il existe une matrice  $\hat{Q}$ . Intéressons nous maintenant au problème d'intégrabilité du  $\mathfrak{R}\left[\frac{d}{dt}\right]$ -idéal  $\Xi$  engendré par les 1-formes  $\omega$  définis par l'équation (II.39). Soit  $z$  un jeu de sorties plates et  $h$  la fonction définie par l'équation (II.21). Les sorties plates vérifient la contrainte  $dz = M\omega$ . Faisons maintenant le choix particulier des sorties plates et de la fonction  $h$  associée :

$$\begin{bmatrix} z_1 \\ z_2 \\ z_3 \end{bmatrix} = \begin{bmatrix} \sigma_1 \\ \sigma_2 \\ \sigma_3 \end{bmatrix} \quad (\text{II.71})$$

Il vient que la différentielle exacte de  $z$  se réécrit :

$$\begin{aligned} dz &= \frac{\partial h}{\partial x} dx + \frac{\partial h}{\partial \dot{x}} d\dot{x} + \dots \\ &= \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix} dx \\ &= \begin{bmatrix} d\sigma_1 \\ d\sigma_2 \\ d\sigma_3 \end{bmatrix} \end{aligned} \quad (\text{II.72})$$

Si nous remarquons que par définition  $d(d\sigma_i) = 0, i = 1, 2, 3$ , alors  $d(dz) = 0$  et par conséquent :

$$d(M\omega) = 0 \quad (\text{II.73})$$

Nous venons ainsi de prouver la forte fermeture du  $\mathfrak{R}\left[\frac{d}{dt}\right]$ -idéal  $\Xi$  engendré le vecteur  $\omega$  correspondant au jeu de sorties plates défini par la suite (cf. section II.4.3). Le modèle dynamique d'un satellite rigide est donc différentiellement plat.  $\square$

Définissons donc maintenant l'application qui relie les variables d'états aux sorties plates.

### II.4.3 Sorties plates et bouclage dynamique

La méthode proposée ci-dessous, qui permet de définir l'application entre les variables d'états et les sorties plates, ne prétend pas être généralisable. La définition de cette application, en plus d'être une nouvelle preuve du lemme 3, va nous permettre de définir tous les profils de guidage en fonction des sorties plates et de leurs dérivées successives. Ce bouclage est au coeur de nos algorithmes de génération de trajectoires car il permettra d'évaluer l'ensemble des contraintes en termes de sorties plates.

Soit la sortie plate candidate  $z$  telle que

$$\begin{bmatrix} z_1 \\ z_2 \\ z_3 \end{bmatrix} = \begin{bmatrix} \sigma_1 \\ \sigma_2 \\ \sigma_3 \end{bmatrix} \quad (\text{II.74})$$

Nous noterons par  $\bar{z}$  le vecteur  $[z, \dot{z}, \dots, z^{(k)}, \dots]$ . Nous obtenons trivialement la première partie du bouclage. En observant l'équation de la cinématique (II.58), les vitesses  $p, q, r$  peuvent

s'exprimer en fonction de la sortie plate en inversant l'équation (II.58). Nous obtenons alors la seconde partie du bouclage :

$$\begin{bmatrix} p \\ q \\ r \end{bmatrix} = \left( \frac{1}{4} \begin{bmatrix} 1 + z_1^2 - z_2^2 - z_3^2 & 2(z_1 z_2 - z_3) & 2(z_1 z_3 + z_2) \\ 2(z_1 z_2 + z_3) & 1 - z_1^2 + z_2^2 - z_3^2 & 2(z_2 z_3 - z_1) \\ 2(z_1 z_3 - z_2) & 2(z_2 z_3 + z_1) & 1 - z_1^2 - z_2^2 + z_3^2 \end{bmatrix} \right)^{-1} \begin{bmatrix} \dot{z}_1 \\ \dot{z}_2 \\ \dot{z}_3 \end{bmatrix} \quad (\text{II.75})$$

La troisième partie du bouclage se déduit de l'équation de conservation du moment cinétique (I.8). Nous obtenons ainsi les entrées de commande (II.64). Ainsi, le moment cinétique interne s'exprime dans  $Ri$  en fonction de la vitesse de rotation et du moment cinétique total. Il vient

$$\begin{bmatrix} H_{int,1} \\ H_{int,2} \\ H_{int,3} \end{bmatrix} = \begin{bmatrix} H_{tot,1} \\ H_{tot,2} \\ H_{tot,3} \end{bmatrix} - \mathbb{I} \begin{bmatrix} p(z) \\ q(z) \\ r(z) \end{bmatrix} \quad (\text{II.76})$$

Enfin, nous pouvons aussi calculer le couple moteur à partir de l'équation (I.8) et donc de l'équation de dynamique du satellite (I.10) :

$$\dot{H}_{tot} = 0 \quad (\text{II.77})$$

$$u = \dot{H}_{int} = -\mathbb{I}\dot{\Omega} - \Omega \wedge H_{tot} \quad (\text{II.78})$$

Ce qui nous donne explicitement :

$$\begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} = -\mathbb{I} \begin{bmatrix} \dot{p}(\bar{z}) \\ \dot{q}(\bar{z}) \\ \dot{r}(\bar{z}) \end{bmatrix} - \begin{bmatrix} p(\bar{z}) \\ q(\bar{z}) \\ r(\bar{z}) \end{bmatrix} \wedge \begin{bmatrix} H_{tot,1} \\ H_{tot,2} \\ H_{tot,3} \end{bmatrix} \quad (\text{II.79})$$

Nous venons de définir les fonctions  $\psi_x$  et  $\psi_u$  telles que nous avons

$$x = \psi_x(\bar{z}) \quad (\text{II.80})$$

$$u = \psi_u(\bar{z}) \quad (\text{II.81})$$

Voyons maintenant comment la platitude différentielle transforme notre problème de génération de trajectoires tel que posé dans le chapitre I.

## II.5 Apports de la platitude au problème de commande optimale

Dans le chapitre I, nous avons établi le problème de génération de trajectoires d'attitude sous contraintes comme un problème de commande optimale. Rappelons que ce problème consiste à déterminer une trajectoire  $t \mapsto (x(t), u(t))$  qui minimise un critère de coût  $J$  et vérifie la dynamique du système, les contraintes transverses (conditions initiales et finales) ainsi que les contraintes de passage et d'actionneurs :

$$\left\{ \begin{array}{l} \dot{x} = f(x, u) \\ x(t_0) = x_0, \quad u(t_0) = u_0 \\ x(t_f) = x_f, \quad u(t_f) = u_f \\ l_b(t) \leq \gamma(x(t), u(t)) \leq u_b(t) \end{array} \right. \quad (\text{II.82})$$

Dans le cas du guidage en attitude, ce problème a été résolu principalement par des méthodes présentées dans le chapitre I, à savoir des méthodes indirectes de tirs et des méthodes directes de transcription dites de collocation.

La platitude différentielle présente de nombreux avantages dans le contexte d'une méthode de collocation. En effet, elle permet d'obtenir une paramétrisation minimale des trajectoires du système (états et commandes). Le problème de commande optimale portera alors sur les variables de cette paramétrisation qui sont appelées sorties plates. De plus, cette paramétrisation permet de vérifier implicitement la dynamique du système : la dynamique de ces sorties plates représentée par une chaîne d'intégrateurs est identiquement équivalente à la dynamique du système non linéaire. Cela aura pour effet d'éliminer toute intégration du processus d'optimisation de trajectoire : le vecteur de contraintes est réduit aux seules contraintes sur la trajectoire. Ainsi, le gain sur la dimension du problème est immédiat vis-à-vis d'une formulation classique des méthodes de collocation. Le problème de guidage va donc se résumer à calculer une trajectoire  $t \mapsto z(t)$ , suffisamment dérivable, qui minimise le critère  $J$  et satisfait les contraintes transverses de passage et d'actionneurs :

$$\begin{cases} \bar{z}(t_i) = \bar{z}_i, & \bar{z}(t_f) = \bar{z}_f \\ l_b(t) \leq \Gamma(\bar{z}(t)) \leq u_b(t) \end{cases} \quad (\text{II.83})$$

Notons que  $\Gamma(\bar{z}(t)) \equiv \gamma(x(t), u(t))$ .

De plus les propriétés de la sortie plate permettent la description de son évolution par tout type de courbes suffisamment dérivables (cf les propriétés du corollaire 1), ce qui nous permettra de choisir le paramétrage des trajectoires des sorties plates.

**Remarque II.3** *Dans le cadre de nos travaux, nous n'envisagerons pas d'associer la platitude avec les méthodes indirectes pour plusieurs raisons. Tout d'abord, en appliquant les méthodes indirectes nous perdons l'avantage de la paramétrisation minimale du problème. En effet, il faudrait alors associer aux sorties plates un jeu d'états adjoints. De plus, cela nous imposerait de connaître la structure du système trivial pour définir la dynamique des états adjoints et faire apparaître les entrées de commande du système trivial. Ensuite, malgré le caractère linéaire du système plat équivalent, l'expression des contraintes en termes des sorties plates s'avère de manière générale complexe, ce qui rendra très difficile les différents calculs variationnels nécessaires à la méthode. Ces difficultés forment malgré tout un problème ouvert qui serait intéressant de résoudre pour savoir si la platitude réduit les différences entre les méthodes directes et indirectes.*

## II.6 Conclusion

Au cours de ce chapitre, nous avons introduit les principales notions qui définissent et caractérisent la platitude. La formulation algébrique de la platitude nous a permis de déterminer les sorties plates qui paramètrent complètement les trajectoires du modèle non linéaire représentant la dynamique d'un satellite rigide. Ceci démontre la platitude de ce modèle. Ces propriétés prennent tout leur sens lorsque l'on déploie le problème de commande optimale. Elles permettent non seulement de réduire le nombre de variables en jeu dans la résolution du problème de commande optimale, mais aussi de choisir la représentation des sorties plates dans l'ensemble des fonctions suffisamment dérivables. Nous allons mettre en place, dans le chapitre suivant, des méthodes de collocation pour systèmes plats et les appliquer au problème de guidage.



---

## Chapitre III

# Génération de trajectoires optimales par platitude et collocation

---

### Sommaire

---

<b>III.1</b>	<b>Introduction</b>	<b>51</b>
<b>III.2</b>	<b>Contexte</b>	<b>51</b>
<b>III.3</b>	<b>Optimisation paramétrique globale</b>	<b>52</b>
III.3.1	Rappel du problème de commande optimale plat	52
III.3.2	Paramétrisation des sorties plates par B-splines	54
III.3.2.1	Principe de collocation par B-splines	54
III.3.2.2	Paramétrisation des sorties plates $z(t)$	54
III.3.3	Problème d'optimisation paramétrique	55
<b>III.4</b>	<b>Optimisation paramétrique sous-optimale</b>	<b>56</b>
III.4.1	Approximation convexe intérieure	57
III.4.1.1	Méthodes d'annexion : approche polytopique	57
III.4.1.2	Méthodes d'annexion superellipsoïdales	58
III.4.1.2.1	Définition des superellipsoïdes	59
III.4.1.2.2	Problème d'approximation	61
III.4.2	Problème d'optimisation convexe	63
III.4.2.1	Critère de coût convexe	64
III.4.2.2	Contraintes quadratiques	64
III.4.2.3	Contraintes linéaires	64
III.4.3	Mesure du conservatisme	65
III.4.3.1	Algorithme d'approximation de $S_{\bar{z}}$	65
III.4.3.1.1	Résolution du problème de satisfaction de contraintes $P_1$	66

III.4.3.1.2	Résolution du problème de satisfaction de contraintes $P_2$ . . . . .	66
III.4.3.1.3	Résolution du problème de satisfaction de contraintes $P_3$ . . . . .	67
III.4.3.1.4	Volume de $S_O$ . . . . .	68
<b>III.5</b>	<b>Etude d'une trajectoire</b> . . . . .	<b>70</b>
III.5.1	Définition de la mission . . . . .	70
III.5.2	Etude des paramètres de configuration . . . . .	71
III.5.2.1	Paramètres de définition des B-splines . . . . .	71
III.5.2.2	Distribution des points de collocation . . . . .	72
III.5.3	Optimisation non convexe . . . . .	73
III.5.4	Optimisation convexe . . . . .	76
III.5.4.1	Contraintes quadratiques . . . . .	76
III.5.4.2	Contraintes linéaires . . . . .	77
III.5.4.3	Contraintes non linéaires . . . . .	78
III.5.5	Comparaison des différents algorithmes . . . . .	80
<b>III.6</b>	<b>Conclusions</b> . . . . .	<b>83</b>

---

## III.1 Introduction

Dans le premier chapitre, nous avons exposé les différentes techniques de génération de trajectoires utilisées pour le guidage satellite, en mettant en lumière les avantages et les inconvénients de chacune d'entre elles. Dans le second chapitre, nous avons présenté le concept de platitude différentielle et ses propriétés.

A ce stade, nous allons proposer une méthode de génération de trajectoires qui associe une technique de collocation directe à la platitude différentielle. La méthode de collocation directe présente l'avantage de résoudre le problème de commande optimale par le biais d'outils de programmation paramétriques non linéaires. La platitude fournit une paramétrisation de l'évolution de l'état et des entrées du système par un jeu de sorties. Nous appliquerons dans ce chapitre cette méthode au problème de guidage, et nous étudierons les possibilités d'améliorer la procédure de génération de trajectoires du point de vue des performances numériques. Dans ce cadre notre contribution sera double. Tout d'abord nous proposons une méthodologie d'approximation convexe de l'espace admissible particulièrement adaptée au cas où l'espace des sorties plates est de grande dimension. Ensuite, nous développons un outil de comparaison des différents schémas d'optimisation qui seront exposés au cours de ce chapitre.

Nous commencerons en rappelant les deux courants de pensées qui exploitent les propriétés de la platitude pour la génération de trajectoires. Nous présenterons, ensuite, une procédure de génération de trajectoires optimales de sorties plates sous les contraintes liées au système. Nous expliquerons comment les méthodes de collocation par B-splines permettent de transformer le problème de commande optimale en un problème d'optimisation paramétrique non linéaire. Ce problème sera considéré comme un problème d'optimisation globale au sens où une solution est recherchée dans l'espace admissible défini par les contraintes considérées.

Ensuite, nous développerons une approche permettant d'utiliser des algorithmes d'optimisation sous optimaux qui allègent la charge de calcul. Le terme "sous optimal" signifie que la solution sera recherchée dans un sous espace convexe de l'espace admissible. Pour compléter notre étude, nous développerons une méthode d'analyse du degré de conservatisme de ces techniques sous optimales.

Enfin, nous proposerons une étude numérique complète ayant pour objet une trajectoire de dépointage d'un satellite d'observation d'orbite basse de type JASON, basé sur la plateforme générique PROTEUS développé conjointement par le CNES et THALES ALENIA SPACE.

## III.2 Contexte

La génération de trajectoires pour les systèmes non linéaires plats a connu un essor important à la fin des années 90. De nombreux exemples académiques et industriels ont été développés par Fliess *et al* [Fliess et al. 1999],[Fliess et al. 1995],[Rouchon 2001]. Une méthode simple pour la prise en compte des contraintes  $y$  est alors donnée : l'évolution de la sortie plate est polynomiale avec des coefficients respectant les conditions aux deux bouts. L'ajout d'une contrainte supplémentaire se traduira par un relâchement de la durée de la trajectoire de manière à la satisfaire. Dans [Lévine & Ngyuen 2003], un exemple complet, allant de la caractérisation des sorties plates à la génération de trajectoires, a été étudié dans le cas d'un système linéaire.

La génération de trajectoires via la résolution du problème de commande optimale par platitude a particulièrement été étudiée au sein de deux équipes de recherche américaines. Agrawal et ses

collaborateurs de l'Université du Delaware ont développé des algorithmes utilisant le cadre de la linéarisation par retour d'état [Agrawal & Faiz 1998, Agrawal, Claewplodtook & Fabien 1998, Faiz & Agrawal 1998]. Ils se sont surtout intéressés à la génération de trajectoires pour systèmes linéarisables par bouclage dynamique. Une vue d'ensemble de cette approche est résumée dans la thèse de Ferreira [Ferreira 2001].

D'autre part, le département Ingénierie et Sciences Appliquée de l'Université CalTech (California Institute of Technology) a développé des outils de génération de trajectoires en temps réel. Le problème de commande optimale est alors résolu par collocation et optimisation non linéaire. Ces travaux, conduits sous la direction de Murray, incluent la réalisation d'un banc de test comportant un turboréacteur orientable. Les premiers travaux ont été réalisés par Nieuwstadt [van Nieuwstadt 1997]. Parmi les contributeurs, on peut compter Milam [Milam 2003] dont la thèse établit la synthèse de cette approche et une application sur le banc du turboréacteur, et les travaux de Petit [Neckel, Talbot & Petit 2003].

Les travaux décrits dans ce chapitre s'inscrivent dans la continuité des travaux précédemment cités. Nous apportons une nouvelle méthode d'approximation convexe de l'espace admissible aux contraintes, et un critère de mesure du conservatisme des solutions qui en résultent.

### III.3 Optimisation paramétrique globale

#### III.3.1 Rappel du problème de commande optimale plat

Le problème de commande optimale a été défini dans la section I.2 pour le système dynamique défini dans le paragraphe II.4.1.2. Rappelons ici les grandes lignes directrices.

Soit le système dynamique d'attitude d'un satellite rigide dont les états  $x$  et les entrées de commande  $u$  sont définis par :

$$x = [\sigma_1, \sigma_2, \sigma_3, p, q, r]^T \quad (\text{III.1})$$

$$u = [\dot{H}_{int,1}, \dot{H}_{int,2}, \dot{H}_{int,3}]^T \quad (\text{III.2})$$

La dynamique du véhicule est modélisée par l'équation suivante :

$$\begin{bmatrix} \dot{\sigma} \\ \dot{\Omega} \end{bmatrix} = \begin{bmatrix} \Theta(\sigma) \\ -\mathbb{I}_{PF}^{-1}(\dot{H}_{int} + \Omega \wedge H_{tot}) \end{bmatrix} \quad (\text{III.3})$$

Nous recherchons la trajectoire  $t \mapsto (x(t), u(t))$  qui minimise le critère suivant :

$$J = \int_{t_0}^{t_f} \dot{H}_{int}^T \dot{H}_{int} dt \quad (\text{III.4})$$

sous les contraintes suivantes :

**Conditions initiales et finales** Les objectifs de la manoeuvre imposent :

$$\begin{cases} x(t_0) = x_0 & u(t_0) = u_0 \\ x(t_f) = x_f & u(t_f) = u_f \end{cases} \quad (\text{III.5})$$

**Contraintes sur la trajectoire** Elles sont liées au système :

1. Les PMR seront contenus dans une sphère unité de dimension 3

$$\sigma_1^2 + \sigma_2^2 + \sigma_3^2 \leq 1 \quad (\text{III.6})$$

2. Les contraintes actionneurs sont définies par les saturations en vitesse et en accélération des roues à réaction :

$$-H_{max} \leq H_{int} \leq H_{max} \quad (\text{III.7})$$

$$-C_{max} \leq \dot{H}_{int} \leq C_{max} \quad (\text{III.8})$$

**Remarque III.1** La contrainte de circonscription des PMR permet d'éviter leur singularité. Cette contrainte implique de limiter l'angle  $\theta$  de la rotation principale  $\mathcal{R}_s$ . Ici, le rayon  $R_s$  de la sphère est fixé à 1 pour les raisons évoquées dans la remarque de la page 43, ainsi nous avons :

$$\tan \frac{\theta_{max}}{4} = R_s = 1 \implies \theta_{max} = \pi \quad (\text{III.9})$$

Dans le cas où l'on souhaite ne pas utiliser les PMR d'aux, il est possible d'augmenter le rayon de la sphère si les objectifs de la manoeuvre le nécessitent :

$$R_s > 1 \implies \theta_{max} > \pi$$

La platitude différentielle permet d'assurer que les trajectoires d'état déduites des sorties plates vérifient les équations différentielles du système. Or, dans les méthodes de collocation classiques, une intégration explicite ou implicite sous forme de contraintes supplémentaires est nécessaire (cf. chapitre I). De plus, les sorties plates, constituant un atlas des trajectoires du système, elles deviennent les variables du problème d'optimisation. Ainsi, nous rechercherons donc la trajectoire  $t \mapsto \bar{z} = (z, \dot{z}, \dots, z^{(k)}, \dots)^T$  qui minimise le critère  $J(\bar{z})$  et satisfait un jeu de contraintes équivalentes à celles des équations (III.5), (III.6), (III.7). Le problème à résoudre revient alors à :

$$\min_{\bar{z}} J(\bar{z})$$

$$\text{sous les contraintes : } \begin{cases} \bar{z}(t_0) = \bar{z}_0 \\ \bar{z}(t_f) = \bar{z}_f \\ \bar{z}(t) \in S_{\bar{z}} \end{cases} \quad (\text{III.10})$$

### Définition III.1 (Espace admissible $S_{\bar{z}}$ )

L'espace admissible  $S_{\bar{z}}$  est un sous ensemble de l'espace des sorties plates,  $O_{\bar{z}}$  qui est défini dans le corollaire 1.  $S_{\bar{z}}$  est défini par

$$S_{\bar{z}} = \{\bar{z} \mid I_b(t) \leq \Xi(\bar{z}(t)) \leq u_b(t)\} \quad (\text{III.11})$$

où  $\Xi$  est donné par l'ensemble des contraintes inégalitaires suivantes :

$$\begin{cases} z_1^2 + z_2^2 + z_3^2 \leq 1 \\ -H_{max} \leq H_{int}(\bar{z}) \leq H_{max} \\ -C_{max} \leq \dot{H}_{int}(\bar{z}) \leq C_{max} \end{cases} \quad (\text{III.12})$$

**Remarque III.2** En développant les inégalités précédentes, il vient que seules  $z, \dot{z}, \ddot{z}$  sont impliquées dans le système III.12. En conséquence, l'espace des sorties plates  $O_{\bar{z}}$ , auquel nous nous référerons, est de dimension  $n_{\bar{z}} = 9$  correspondant aux coordonnées des composantes de  $z, \dot{z}, \ddot{z}$ . Par la suite, nous utiliserons  $\bar{z}$  en référence au vecteur  $(z_1, z_2, z_3, \dot{z}_1, \dot{z}_2, \dot{z}_3, \ddot{z}_1, \ddot{z}_2, \ddot{z}_3)^T$ .

Notons que dans ce cadre, les contraintes sur la trajectoire et les actionneurs, linéaires et convexes en  $x$  et  $u$ , sont transformées en contraintes non linéaires et généralement non convexes en  $z, \dot{z}, \ddot{z}$ .

### III.3.2 Paramétrisation des sorties plates par B-splines

#### III.3.2.1 Principe de collocation par B-splines

De manière générale, les méthodes directes de transcription consistent à décrire les commandes et/ou les états dans un espace fini de fonctions :

$$u \in \text{span}\{\hat{u}_1, \dots, \hat{u}_{n_u}\}, \quad u = \sum_{i=1}^{n_u} \alpha_i \hat{u}_i \quad (\text{III.13})$$

$$x \in \text{span}\{\hat{x}_1, \dots, \hat{x}_{n_x}\}, \quad x = \sum_{j=1}^{n_x} \beta_j \hat{x}_j \quad (\text{III.14})$$

Rappelons que la collocation consiste à rechercher, dans ces espaces de fonction, une solution aux équations différentielles associées à la dynamique du système. Les polynômes par morceaux, dit splines, ont très souvent été pressentis pour résoudre ce problème d'intégration implicite [Ascher, Press & Russell 1983], notamment lorsque celui-ci est inclus dans un problème de commande optimale [Renes 1978, Kraft 1985, Hargraves & Paris 1987]. Plus précisément, les B-splines ont fréquemment été étudiés pour approcher les solutions de systèmes d'équations différentielles ordinaires (cf [Swartz 1988]). Ce choix peut notamment s'expliquer par le travail fondamental de de Boor [de Boor 1978], qui a fourni un nombre important de résultats théoriques ainsi que des algorithmes robustes de calcul pour les B-splines.

#### III.3.2.2 Paramétrisation des sorties plates $z(t)$

Ici, les sorties plates sont paramétrées à la place des états et des commandes. L'intérêt des B-splines pour la paramétrisation des sorties plates est multiple. Il est relativement simple d'établir la classe de continuité sur la trajectoire avec un nombre raisonnable de variables et sans contrainte supplémentaire. De plus, les B-splines sont à support local : chaque élément est non nul uniquement localement, ce qui apporte de la stabilité numérique aux techniques qui vont être employées par la suite. Ainsi, les sorties plates vont se définir de la manière suivante :

$$z_i(t) = \sum_j^n C_{i,j} \cdot B_{j,k}(t), \quad i = 1, 2, 3 \quad (\text{III.15})$$

où  $C_{i,j}$  sont les coefficients aussi appelés points de contrôle et  $B_{j,k}$  les B-splines d'ordre  $k$  construites sur la séquence  $T$  strictement non décroissante de points temporels  $\{t_i\}$  (cf annexe

C). Les trajectoires des sorties plates définies par des B-splines sont dérivables sur l'ouvert  $]t_0, t_f[$  :

$$z_i^{(s)}(t) = \sum_j^n C_{i,j} \cdot B_{j,k}^{(s)}(t), \quad i = 1, 2, 3 \quad (\text{III.16})$$

**Remarque III.3** Bien que les B-splines soient particulièrement bien adaptées à notre problème, d'autres choix sont possibles notamment les séries de Fourier ou les polynômes orthogonaux. Les polynômes orthogonaux nécessitent néanmoins un ordre élevé pour répondre à des contraintes complexes, tandis que les séries de Fourier sont plus adaptées aux trajectoires périodiques.

### III.3.3 Problème d'optimisation paramétrique

Rappelons que le problème de commande optimale en termes de sorties plates est donné par (III.10). La paramétrisation des sorties plates par des B-splines permet de transformer le problème (III.10) de dimension infinie en un problème de programmation semi infinie suivant :

$$\begin{aligned} & \min_C J(C) \\ & \text{sous les contraintes : } \begin{cases} \bar{z}(t_i, C) = \bar{z}_i \\ \bar{z}(t_f, C) = \bar{z}_f \\ \bar{z}(C) \in S_{\bar{z}} \end{cases} \quad (\text{III.17}) \end{aligned}$$

où  $C = (C_{1,1}, \dots, C_{1,n}, C_{2,1}, \dots, C_{2,n}, C_{3,1}, \dots, C_{3,n})$  est le vecteur des coefficients définissant les trajectoires (III.15). Le problème (III.17) est semi-infinie car les variables d'optimisation sont en nombre fini mais elles doivent vérifier les contraintes sur la trajectoire et les actionneurs sur un continuum temporel.

La méthode de collocation se base alors sur une méthode de discrétisation des contraintes : les contraintes seront vérifiées en un nombre fini de points  $\{\tau_i\}$  appelés points de collocation par référence aux points où la dynamique du système est vérifiée dans les méthodes classiques de collocation. Le quadrillage de collocation (i.e. nombre et distribution des points) est un degré de liberté pour la procédure de génération de trajectoires.

Finalement, nous obtenons le problème d'optimisation paramétrique non linéaire (ou *non linear programming NLP*) suivant :

$$\begin{aligned} & \min_C J(C) \\ & \text{sous les contraintes : } \begin{cases} \bar{z}(t_i, C) = \bar{z}_i \\ \bar{z}(t_f, C) = \bar{z}_f \\ \bar{z}(C, \tau_i) \in S_{\bar{z}}, \quad i = 1, 2, \dots, \text{card}\{\tau_i\} \end{cases} \quad (\text{III.18}) \end{aligned}$$

En résumé, pour configurer ce problème il est nécessaire de définir un certain nombre de paramètres :

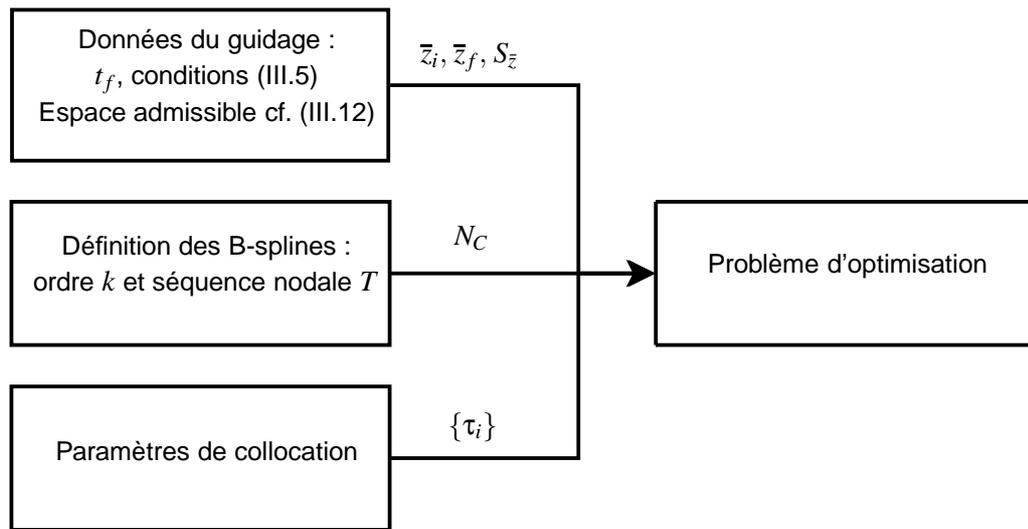


FIG. III.1 – Schéma de configuration du problème d'optimisation

- **Données du problème de guidage** Le temps de manoeuvre et les conditions initiales et finales (à savoir l'attitude, la vitesse, l'accélération et si besoin le jerk) vont déterminer  $\bar{z}_i$  et  $\bar{z}_f$  à partir des fonctions  $\psi_x$  et  $\psi_u$  (cf chapitre II paragraphe II.4.3). Les contraintes sur la trajectoire des sorties sont définies à partir du système d'inégalités (III.12).
- **Paramétrisation des sorties plates** Les B-splines se définissent à partir de l'ordre  $k$  et de la séquence nodale  $T$ . Ce choix déterminera la dimension du vecteur  $C$ .
- **Paramètres de collocation** Le choix du nombre et de la distribution des points de collocation nécessite une certaine expertise dans le domaine. Il détermine la taille du problème NLP et donc sa complexité.

La taille du problème dépend à la fois du nombre des variables à optimiser (i.e. le nombre d'éléments du vecteur  $C$ ) et du nombre de points de collocation  $\tau_i$ . Chaque inégalité du système (III.12) implique une contrainte non linéaire vérifiée en chaque point de  $\{\tau_i\}$  et dont la complexité est inhérente au bouclage dynamique (cf chapitre II paragraphe II.4.3). D'un point de vue algorithmique, il pourra être intéressant de remplacer le système de contraintes (III.12) par un système de contraintes convexes. Pour ce faire, nous allons remplacer, dans le problème (III.18), l'espace admissible  $S_{\bar{z}}$  par un sous-espace convexe inscrit dans ce dernier.

### III.4 Optimisation paramétrique sous-optimale

Afin d'accroître l'efficacité de l'optimisation vis-à-vis de l'évaluation des contraintes (III.12), nous allons établir une procédure qui mettra en place des contraintes convexes en termes de  $\bar{z}$ . Pour ce faire, nous allons définir des approximations convexes intérieures  $S_I$  qui remplaceront l'espace admissible initial dans le problème d'optimisation paramétrique (III.18). Ce problème d'optimisation est sous optimal au sens où l'espace de recherche d'une solution est un sous-espace de l'espace admissible.

Dans un premier temps nous allons développer deux méthodes d'approximation de  $S_{\bar{z}}$  qui permettront de fournir des sous-espaces convexes admissibles vis-à-vis du système de contraintes (III.12). Dans un second temps, nous ferons une étude volumétrique de l'espace admissible

$S_{\bar{z}}$  qui permettra par la suite d'établir un critère de mesure du conservatisme des algorithmes utilisant ces approximations.

### III.4.1 Approximation convexe intérieure

Le but de ce paragraphe est d'approcher  $S_{\bar{z}}$  par un sous-espace convexe  $S_I$  de l'espace des sorties plates tel que  $S_I \subset S_{\bar{z}}$ . Il existe dans la littérature deux types de méthodes : une méthode analytique et des méthodes numériques. Les méthodes numériques ont l'avantage de s'adapter aux environnements plus complexes (i.e. modèles du système, bouclage dynamique, contraintes), tandis que la méthode analytique présente l'intérêt d'être basée sur des conditions nécessaires et suffisantes et fournissent une solution analytique du problème. Faiz [Faiz 1999] propose deux visions assez différentes pour obtenir des sous-ensembles convexes. La première est une recherche analytique élégante de l'approximation convexe intérieure. Elle consiste à chercher une zone atteignable de l'espace d'état du système (i.e.  $(x, u)$ ) telle que l'image de cette zone par le bouclage dynamique soit convexe. Il établit alors une condition nécessaire et suffisante que doivent vérifier les zones atteignables de l'espace d'état. Malgré l'élégance mathématique, elle s'avère peu pratique à utiliser dans le cas d'un bouclage dynamique complexe et de  $n_{\bar{z}}$  grand. La seconde approche est une approche numérique et polytopique développée dans [Faiz et al. 2001]. Nous présenterons ici l'approche polytopique de Faiz et une méthode d'annexion d'espace par superellipsoïdes.

#### III.4.1.1 Méthodes d'annexion : approche polytopique

La philosophie de la démarche est assez simple : elle consiste à choisir un premier sous-espace  $S_I$  inclus dans  $S_{\bar{z}}$  et de l'élargir tant qu'il reste inscrit dans  $S_{\bar{z}}$ . Ces méthodes reposent essentiellement sur des méthodes numériques de recherche. Les résultats de Faiz en la matière [Faiz 1999] se situent dans la continuité de différentes approches qui ont été développées depuis les années 50. On peut notamment citer les travaux de Horst et Tuy [Horst & Tuy 1996], qui, pour obtenir une approximation intérieure convexe, se basent sur l'approximation intérieure d'espaces admissibles convexes par des polytopes. Les travaux de Fliesher et al [Fliesher, Mehlorn, Rote, Welzl & Yap 1992] proposent, quant à eux, un algorithme de détermination du plus grand polytope convexe intérieur et du plus petit polytope convexe extérieur à nombre de sommets fixé. Le même type d'algorithme a été proposé par Boyd et al avec des ellipsoïdes [Boyd, El Ghaoui, Feron & Balakrishnan 1994].

Faiz propose une méthode permettant d'approcher des domaines non convexes. Le principe consiste à maximiser le volume d'un polytope sous la contrainte qu'il reste inclus dans le domaine admissible. Il pose alors un problème d'optimisation pour trouver le plus grand polytope intérieur au domaine accessible.

Soit  $\mathcal{P}_{n_{\bar{z}}}$  un polytope de  $O_{\bar{z}}$  défini par l'ensemble de ces sommets  $V = \{V_1, V_2, \dots, V_{q_v}\}$  et on note  $F$  l'ensemble de ses facettes. Tout point  $\bar{z}$  du polytope  $\mathcal{P}_{n_{\bar{z}}}$  est défini par le vecteur  $\lambda = (\lambda_1, \lambda_2, \dots, \lambda_{q_v})^T$  tel que :

$$\bar{z} = (V, \lambda) = \lambda_1 V_1 + \lambda_2 V_2 + \dots + \lambda_{q_v} V_{q_v}, \lambda_j \in [0, 1], \sum_{j=1}^{q_v} \lambda_j = 1 \quad (\text{III.19})$$

Notons par  $\Xi(V, \lambda)$  la fonction d'évaluation des contraintes inégalitaires (III.12) sur les points du polytope. Le problème d'approximation polytopique se pose alors en termes d'optimisation

paramétrique :

$$\max_{V \in D} \text{Volume} \mathcal{P}_{n_{\bar{z}}}(V) \quad (III.20)$$

$$\text{avec} \begin{cases} D = \{V | \Xi(V, \lambda) \leq 0, \lambda \in B\} \\ B = \{\lambda \in \mathbb{R}^{q_v} | \lambda_j \in [0, 1], \sum_{j=1}^{q_v} \lambda_j = 1\} \end{cases}$$

Le critère à optimiser est donc le volume du polytope. Les variables d'optimisation sont les coordonnées des sommets du polytope. La contrainte d'inscription dans le domaine admissible est à vérifier sur l'ensemble des facettes i.e. sur le continuum des paramètres  $\lambda_j$ . Cette description évoque un problème d'optimisation semi-infinie. L'utilisation d'une méthode de réduction locale des contraintes permet alors de définir le problème d'optimisation en termes de dimension finie [Hettich & Kortanek 1993]. Ces techniques de réduction locale spécifient les conditions d'équivalence entre le problème d'optimisation semi-infinie originel et le problème d'optimisation paramétrique discrétisé. Ainsi, les contraintes sont évaluées en un nombre fini de points du polytope. On procède par la suite au quadrillage des facettes du polytope où seront vérifiées les contraintes. On pose alors le problème suivant :

$$\max_{V \in D} \text{Volume} \mathcal{P}_{n_{\bar{z}}}(V) \quad (III.21)$$

$$\text{avec} \begin{cases} D = \{V | \Xi(V, \lambda) \leq 0, \lambda \in \tilde{B}\} \\ \tilde{B} = \{\{\lambda_1, \lambda_2, \dots, \lambda_{q_\lambda} \in B | \bar{z} \in F\} \end{cases}$$

Faiz [Faiz et al. 2001] propose une procédure qui peut être résumée comme suit :

1. Définir le nombre de sommets et facettes
2. Création d'un polytope primitif
3. Création du quadrillage d'évaluation des contraintes par méthode de réduction locale
4. Résolution du problème d'optimisation (III.21)

La création d'un polytope primitif soulève un certain nombre de difficultés. La définition d'un polytope nécessite un nombre de facettes et de sommets fixés. Or, il en existe une très grande variété. Faiz [Faiz et al. 2001] propose alors d'inscrire ce polytope primitif dans une sphère unité. L'inconvénient d'une méthode qui consiste à optimiser les coordonnées des sommets d'un polytope, est que celles-ci vont évoluer, de façon mal maîtrisée, au cours du processus d'optimisation. Or, la complexité du problème augmente exponentiellement avec le nombre de sommets, qui, croît, à son tour, fortement avec la dimension de  $O_{\bar{z}}$ . Faiz [Faiz 1999] propose plusieurs exemples d'approximations polytopiques réussies dans le cas d'espace de sorties plates de dimension 2. Or, dans notre cas, la dimension de  $O_{\bar{z}}$  est 9 et le bouclage dynamique est complexe. Il devient alors difficile de mettre en place de telles techniques. Dans les paragraphes suivants, nous proposons une méthode pour gérer convenablement le problème de la grande dimension de l'espace des sorties plates et du bouclage dynamique complexe.

#### III.4.1.2 Méthodes d'annexion superellipsoidales

L'approche que nous proposons ici a pour but de simplifier le processus d'approximation convexe, notamment en limitant le nombre de paramètres d'entrée. Notre approche se base sur

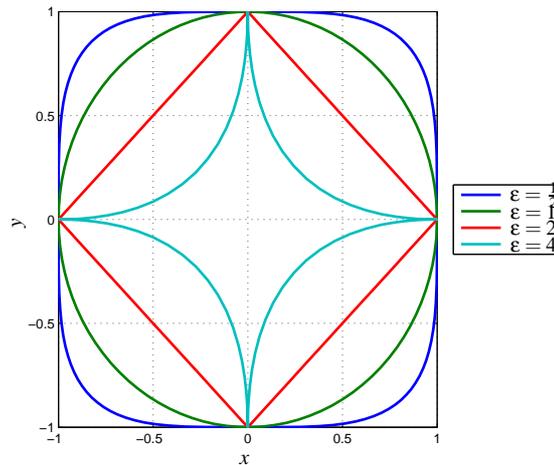


FIG. III.2 – Variation de la superellipse en fonction de  $\varepsilon$

les surfaces dites superellipsoïdales. Tout d'abord, nous donnerons la définition des superellipsoïdes et les propriétés géométriques utiles. Ensuite, nous exposerons le problème d'approximation et nous montrerons que notre approche, bien qu'équivalente en certains points à celle de Faiz, permet de réduire le nombre des paramètres d'optimisation à  $n_{\bar{z}}$  ( $n_{\bar{z}} \times q_v$  pour l'approche de Faiz [Faiz et al. 2001]).

**III.4.1.2.1 Définition des superellipsoïdes** Les superellipsoïdes peuvent être vues comme une extension des superellipses en dimension  $n$  [Barr 1981].

**Définition III.2 (Superellipse)**

Une superellipse est une courbe fermée du plan  $(O, x, y)$  définie par l'équation suivante :

$$\left(\frac{x}{e_1}\right)^{\frac{2}{\varepsilon}} + \left(\frac{y}{e_2}\right)^{\frac{2}{\varepsilon}} = 1 \tag{III.22}$$

$e_1$  et  $e_2$  étant les demi-grands axes et  $\varepsilon \in \mathbb{R}^{*,+}$ . Notons que, pour le calcul, les termes  $\frac{x}{e_1}$  et  $\frac{y}{e_2}$  sont d'abord élevés au carré.

Pour  $e_1 = e_2$  et  $\varepsilon = 1$ , nous obtenons le cercle centré en 0 de rayon  $e_1$ . Lorsque  $\varepsilon$  tend vers 0 la superellipse tend vers le carré de côté parallèle aux axes et de longueur 1 et lorsque  $\varepsilon$  augmente, la superellipse tend vers une étoile à quatre branches (cf. figure III.2).

La courbe d'une superellipse se réécrit en coordonnées polaires :

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} e_1 \cos^\varepsilon \theta \\ e_2 \sin^\varepsilon \theta \end{bmatrix}, \quad -\pi \leq \theta \leq \pi \tag{III.23}$$

On appelle  $\theta$  l'anomalie et  $\varepsilon$  l'exposant de la superellipse [Barr 1981]. Par convention les puissances de fonctions trigonométriques se calculent de la façon suivante :

$$\begin{aligned} \cos^\varepsilon \theta &= \text{sign}(\cos \theta) |\cos \theta|^\varepsilon \\ \sin^\varepsilon \theta &= \text{sign}(\sin \theta) |\sin \theta|^\varepsilon \end{aligned}$$

Les superellipsoïdes en dimension 3 s'obtiennent alors par le produit sphérique entre deux superellipses :

$$\begin{aligned} \begin{bmatrix} x \\ y \\ z \end{bmatrix} &= \begin{bmatrix} e_{1,1} \cos^{\varepsilon_1} \theta_1 \\ e_{1,2} \sin^{\varepsilon_1} \theta_1 \end{bmatrix} \otimes \begin{bmatrix} e_{2,1} \cos^{\varepsilon_2} \theta_2 \\ e_{2,2} \sin^{\varepsilon_2} \theta_2 \end{bmatrix} \\ &= \begin{bmatrix} e_{1,1} e_{2,1} \cos^{\varepsilon_2} \theta_2 \cos^{\varepsilon_1} \theta_1 \\ e_{1,1} e_{2,2} \cos^{\varepsilon_2} \theta_2 \sin^{\varepsilon_1} \theta_1 \\ e_{1,2} \sin^{\varepsilon_2} \theta_2 \end{bmatrix}, \quad \begin{cases} -\pi \leq \theta_1 \leq \pi \\ -\frac{\pi}{2} \leq \theta_2 \leq \frac{\pi}{2} \end{cases} \end{aligned} \quad (\text{III.24})$$

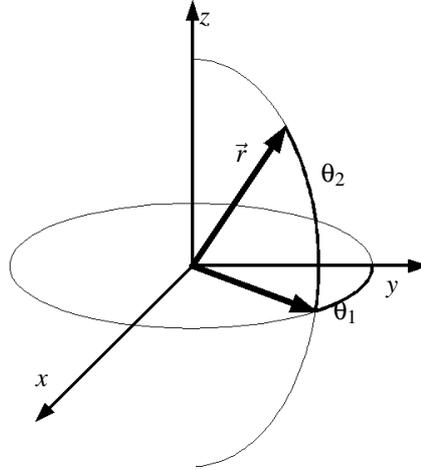


FIG. III.3 – Vecteur définissant une superellipsoïde comme produit sphérique de deux superellipses

Nous pouvons observer sur la figure III.3, que  $\varepsilon_2$  définit la forme de la superellipsoïde sur les plans parallèles à  $(x, y)$  et  $\varepsilon_1$  sur les plans perpendiculaires à  $(x, y)$ . De manière générale, on admettra que pour assurer la convexité de la surface, on devra avoir  $\forall i, \varepsilon_i \leq 2$  [Barr 1981]. On obtient une superellipsoïde  $\mathcal{S}_n$  de dimension  $n$ , en considérant  $n - 1$  superellipses  $E_i$  définies par :

$$E_i(\varepsilon_i, u_i) = \begin{bmatrix} e_{i,1} \cos^{\varepsilon_i} \theta_i \\ e_{i,2} \sin^{\varepsilon_i} \theta_i \end{bmatrix}, \quad e_{i,1}, e_{i,2}, \varepsilon_i \in \mathbb{R}^{+*}, \quad \theta_i \in \begin{cases} [-\pi, \pi] & \text{si } i = 1 \\ [-\frac{\pi}{2}, \frac{\pi}{2}] & \text{si } i = 2, \dots, n-1 \end{cases} \quad (\text{III.25})$$

On notera par la suite par  $\theta = [\theta_1, \dots, \theta_{n-1}]$  le vecteur des anomalies et par  $\varepsilon = [\varepsilon_1, \dots, \varepsilon_{n-1}]$  le vecteur des exposants. Les coordonnées  $[x_1, \dots, x_n]^T$  d'un point de la surface sont données par le produit sphérique des  $n - 1$  superellipses  $E_{n-1} \otimes E_{n-2} \otimes \dots \otimes E_1$  :

$$x_i = \begin{cases} a_i \prod_{k=1}^{n-1} \cos^{\varepsilon_k} \theta_k & \text{si } i = 1 \\ a_i \sin^{\varepsilon_{i-1}} \theta_{i-1} \prod_{k=i}^{n-1} \cos^{\varepsilon_k} \theta_k & \text{si } i = 2, \dots, n \end{cases} \quad (\text{III.26})$$

où  $a_i = f(e_{1,1}, \dots, e_{n-1,2})$ . On note  $a = [a_1, \dots, a_n]$  le vecteur des demi-grands axes. Ainsi, pour définir une superellipsoïde,  $2n - 1$  paramètres sont nécessaires.

Les superellipsoïdes sont des surfaces qui sont plus flexibles que les ellipsoïdes utilisées dans

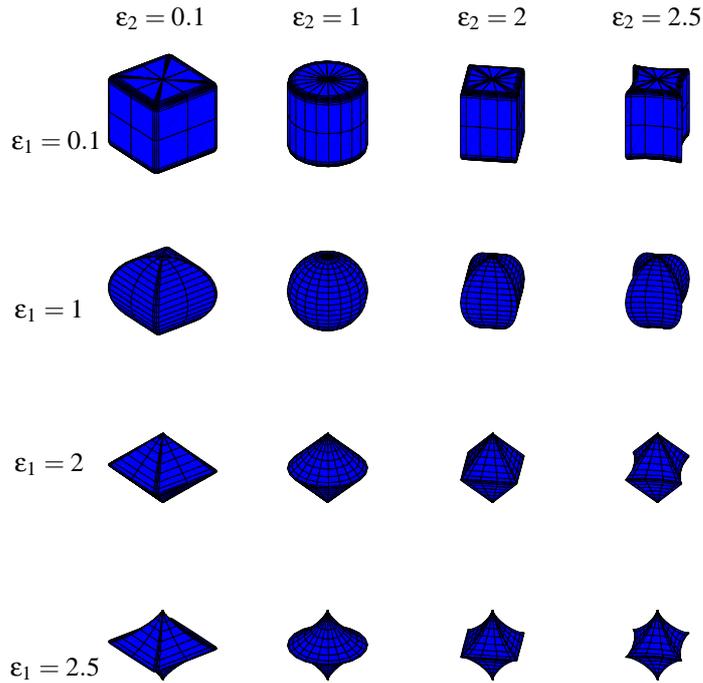


FIG. III.4 – Superellipsoïdes 3D pour différentes valeurs de  $\epsilon_1$  et  $\epsilon_2$

[Boyd et al. 1994] pour les approximations convexes. La convexité de la superellipsoïde est déterminée par les valeurs des paramètres  $\{\epsilon_i\}$  (cf figure III.4 ). En effet, ils permettent aussi de contrôler le caractère anguleux de la surface en fonction de chaque axe.

A l’instar de l’approche polytopique, le problème d’approximation par superellipsoïdes se pose comme l’optimisation du volume décrit par la surface sous la contrainte que ce volume soit inscrit dans l’espace admissible. La définition du volume d’une superellipsoïde en  $n$  dimensions et de la fonction d’appartenance d’un point à la superellipsoïde sont données en annexe D.

**III.4.1.2.2 Problème d’approximation** Le problème d’approximation consiste à maximiser le volume  $V$  de  $S_n$  une superellipsoïde en dimension  $n$  sous la contrainte que la superellipsoïde reste inscrite dans  $S_{\bar{z}}$ . A l’instar de l’approche polytopique, ce problème se pose comme un problème d’optimisation semi-infinie :

$$\max_{a, \epsilon} \text{Volume } \mathcal{S}(a, \epsilon) \text{ sous la contrainte : } \forall \bar{z} \text{ tel que } \mathcal{F}(\bar{z}) \leq 1, \bar{z} \in S_{\bar{z}} \quad (\text{III.27})$$

Le volume de  $S_n$  est défini par :

$$V_n = 2a_n \prod_{i=1}^{n-1} a_i \epsilon_i B\left(\frac{\epsilon_i}{2}, \frac{i}{2} \epsilon_1 + 1\right) \quad (\text{III.28})$$

La fonction d’appartenance  $\mathcal{F}$  est donnée par l’expression suivante :

$$\mathcal{F}_n(x, a) = \Lambda_{n,n}, \quad x, a \in \mathbb{R}^n \quad (\text{III.29})$$

où  $\Lambda_{n,n}$  est définie par la récurrence suivante :

$$\begin{cases} \Lambda_{n,2}(x) = \left(\frac{x_1}{a_1}\right)^{\frac{2}{\varepsilon_1}} + \left(\frac{x_2}{a_2}\right)^{\frac{2}{\varepsilon_1}} \\ \Lambda_{n,k}(x) = (\Lambda_{n,k-1})^{\frac{\varepsilon_{k-2}}{\varepsilon_{k-1}}} + \left(\frac{x_k}{a_k}\right)^{\frac{2}{\varepsilon_{k-1}}} \end{cases} \quad (\text{III.30})$$

Les calculs exprimés ci-dessus sont détaillées dans l'annexe D.

**Remarque III.4** Nous cherchons à maximiser le volume d'une superellipsoïde qui est centrée en l'origine de l'espace des sorties plates  $O_{\bar{z}}$  et dont les demi-grands axes sont alignés sur les axes de  $O_{\bar{z}}$ . Ces faits se justifient par les informations obtenues à partir du système d'inégalités (III.12) qui définit  $S_{\bar{z}}$  :

- $S_{\bar{z}}$  est centré autour de l'origine de  $O_{\bar{z}}$ . En effet, si un point  $\bar{z}$  appartient à  $S_{\bar{z}}$  alors nous avons  $-\bar{z} \in S_{\bar{z}}$ .
- $S_{\bar{z}}$  admet chaque axe de  $O_{\bar{z}}$  comme axe de symétrie. Ces symétries se démontrent par étude de la parité de la fonction  $\Xi$  autour de chaque axe. Cette étude consiste à comparer, pour l'axe  $j = 1, \dots, n_{\bar{z}}$ ,  $\Xi(\bar{z}_a)$  et  $\Xi(\bar{z}_b)$  où

$$\bar{z}_{a,i} = \begin{cases} \bar{z}_{b,i}, & i = j \\ -\bar{z}_{b,i}, & i \neq j \end{cases}$$

Les paramètres d'optimisation sont les exposants  $\varepsilon_i$  et les demi-grands axes  $a_i$ . Les paramètres  $\varepsilon_i$  seront considérés comme des paramètres de configuration du problème car ils définissent la forme de la superellipsoïde. Nous profiterons de la définition de  $S_{\bar{z}}$  (cf. (III.12)) pour fixer une partie des exposants  $\varepsilon_i$ . En remarquant que les trois premières coordonnées de  $\bar{z}$  appartiennent à la sphère unité (cf. la première inégalité du système (III.12)), il apparaît intéressant de fixer la valeur de  $\varepsilon_1$  et  $\varepsilon_2$  à 1. Pour des raisons de simplicité, les valeurs des autres exposants seront choisies égales entre elles :

$$\varepsilon_i = \begin{cases} 1 & \text{if } i = 1, 2 \\ \alpha & \text{if } i = 3, \dots, n-1 \end{cases} \quad (\text{III.31})$$

On se fixe une série de valeurs de  $\alpha \in ]0, 2]$  et pour chaque valeur de  $\alpha$ , les demi-grands axes seront les seules variables d'optimisation. Cette limitation se justifie par la difficulté d'optimiser à la fois la forme et la taille de la superellipsoïde. En effet, la dépendance des paramètres  $a$  vis-à-vis de  $\varepsilon$  peut induire des effets néfastes sur l'algorithme d'optimisation.

Le problème d'approximation (III.32) est un problème de programmation non linéaire semi-infinie. Afin de réduire la taille du problème, nous allons vérifier la contrainte d'appartenance à  $S_{\bar{z}}$  uniquement à la surface de la superellipsoïde en un nombre de points finis.

$$\max_{a,} \text{Volume } \mathcal{S}(a) \text{ sous la contrainte : } \forall \bar{z}(\varepsilon, a, u) \text{ défini par (III.26) , } \bar{z} \in S_{\bar{z}} \quad (\text{III.32})$$

Le quadrillage d'évaluation de la contrainte d'appartenance à  $S_{\bar{z}}$  est défini sur les coordonnées (III.26) où les anomalies  $\theta_i$  prennent des valeurs discrètes sur leurs intervalles de définition :

$$\{\theta_{i,1}, \dots, \theta_{i,q_{\theta_i}}\} \in \begin{cases} [-\pi, \pi] & \text{si } i = 1 \\ [-\frac{\pi}{2}, \frac{\pi}{2}] & \text{si } i = 2, \dots, n_{\bar{z}} - 1 \end{cases} \quad (\text{III.33})$$

Notre méthode d'approximation peut se résumer par la démarche suivante :

1. Définir les  $n_{\bar{z}}$  exposants  $\varepsilon_i$  (cf. (III.31))
2. Création du quadrillage d'évaluation des contraintes
3. Résolution du problème d'optimisation (III.32)

Pour la plupart des exemples de superellipsoïdes calculés sur le tableau III.1, les exposants sont définis par la formule (III.31). Les exposants de la dernière superellipsoïde sont tous égaux à 2 ce qui permet d'obtenir un cas particulier dont nous exploiterons les propriétés dans le paragraphe III.4.2.3. Le problème d'optimisation (III.32) a été résolu avec un quadrillage fin d'évaluation i.e. nous avons pris 4 valeurs discrètes par anomalie. On obtient alors  $4^8$  points ce qui après éliminations des points doubles nous ramène à environ 14000 points d'évaluation des contraintes. L'application de la procédure d'approximation donne les résultats du tableau III.1. On constate que le volume diminue avec l'accroissement du paramètre  $\alpha$ .

	$S_1 \diamond$	$S_2 \diamond$	$S_3 \diamond$	$S_4 \diamond$	$S_5 \diamond$	$S_6 \sharp$
$\alpha$	0.1	0.5	1	1.5	2	2
volume	1.0e-11	2.24e-12	1.38e-13	5.63e-15	1.85e-16	5.89e-17

TAB. III.1 – Calcul des superellipsoïdes ( $\diamond$  :  $\varepsilon_i$  définis par (III.31),  $\sharp$  :  $\varepsilon_i = \alpha, i = 1, \dots, n_{\bar{z}} - 1$ )

### III.4.2 Problème d'optimisation convexe

Le problème d'optimisation paramétrique qui sera résolu diffère du problème décrit dans le paragraphe III.3.3. En effet, nous remplacerons  $S_{\bar{z}}$  par les approximations convexes intérieures  $S_I$  que nous avons présentées dans cette section et nous définirons un critère convexe :

$$\min_C J_C(C)$$

$$\text{sous les contraintes : } \begin{cases} \bar{z}(t_i, C) = \bar{z}_i \\ \bar{z}(t_f, C) = \bar{z}_f \\ \bar{z}(C, \tau_i) \in S_I, \quad i = 1, 2, \dots, \text{card}\{\tau_i\} \end{cases} \quad (\text{III.34})$$

Dans le cas des superellipsoïdes, la contrainte sur la trajectoire s'exprime à partir de la fonction d'appartenance définie pour une superellipsoïde de dimension  $n$  :

$$\bar{z}(C, \tau_i) \in S_I \Leftrightarrow \mathcal{F}_{n_{\bar{z}}}(\bar{z}(C, \tau_i)) \leq 1, \quad i = 1, 2, \dots, \text{card}\{\tau_i\} \quad (\text{III.35})$$

Le principal avantage de cette formulation, vis-à-vis des contraintes (III.12), est de réduire le nombre de contraintes inégalitaires de 13 pour le système (III.12) à 1 pour le problème (III.34). De plus, les fonctions  $\psi_x$  et  $\psi_u$  ne sont plus évaluées car les contraintes portent désormais sur les sorties plates et leurs dérivées.

La génération d'un problème d'optimisation convexe passe aussi par la définition d'un critère de coût convexe vis-à-vis des paramètres d'optimisation. Nous proposons dans les paragraphes suivants un critère convexe de coût et des contraintes linéaires et quadratiques qui peuvent se déduire facilement des superellipsoïdes étudiées précédemment.

### III.4.2.1 Critère de coût convexe

Il paraît intéressant d'intégrer l'accélération des sorties plates  $\ddot{z}$  dans la fonction de coût. En effet,  $\ddot{z}$  est linéairement dépendant de  $C$  car nous avons :

$$\ddot{z}_i(t) = \sum_j^n C_{i,j} \cdot \ddot{B}_{j,k}(t), \quad i = 1, 2, 3 \quad (\text{III.36})$$

Ainsi, notre critère de coût convexe est donné par :

$$J_C = \int_{t_0}^{t_f} \ddot{z}^T \ddot{z} dt \quad (\text{III.37})$$

### III.4.2.2 Contraintes quadratiques

Les contraintes d'appartenance sont le plus généralement non linéaires mais il est possible d'obtenir des contraintes de type quadratiques ou linéaires pour des cas particuliers de superellipsoïdes comme les ellipsoïdes. Elles sont obtenues en posant  $\varepsilon_i = 1$ ,  $i = 1, \dots, n_{\bar{z}} - 1$  ce qui correspond à la superellipsoïde  $S_3$ . Une ellipsoïde peut aussi être définie par une équation matricielle quadratique :

$$\mathcal{E}_n = \{x | (x - x_c)^T P (x - x_c) = 1\} \quad (\text{III.38})$$

où la matrice  $P$  est une matrice symétrique. Dans le cas où la direction des demi-grands axes est confondue avec les axes du repère,  $P$  est une matrice diagonale dont la diagonale contient les composantes du vecteur  $a^1$ . Nous venons de montrer qu'il est aussi possible d'utiliser les ellipsoïdes pour approcher des régions non convexes. Ainsi, les contraintes sur la trajectoire s'exprimeront de la façon suivante :

$$\bar{z}(C, \tau_i) \in \mathcal{E}_{n_{\bar{z}}} \Leftrightarrow \bar{z}^T P \bar{z} \leq 1, \quad i = 1, 2, \dots, \text{card}\{\tau_i\} \quad (\text{III.39})$$

Notons que nous pouvons déduire des résultats de la remarque III.4 que  $P$  est une matrice diagonale formée à partir du vecteur des demi-grands axes.

### III.4.2.3 Contraintes linéaires

De même il est possible de déduire des contraintes linéaires à partir d'un cas particulier de superellipsoïdes. La surface  $S_6$  est ce cas particulier obtenu pour

$$\varepsilon_i = 2, \quad i = 1, \dots, n_{\bar{z}} - 1$$

Elle correspond à un polytope dont les sommets sont situés sur les axes du repère. Les coordonnées des  $2n_{\bar{z}}$  sommets sont donnés par la valeur des demi-grands axes :

$$V_{i,j} = \begin{cases} 0 & \text{si } i \neq j \\ a_i & \text{si } i = j \end{cases}, \quad i, j = 1, \dots, n_{\bar{z}} \quad (\text{III.40})$$

$$V_{i+n_{\bar{z}},j} = \begin{cases} 0 & \text{si } i \neq j \\ -a_i & \text{si } i = j \end{cases}, \quad i, j = 1, \dots, n_{\bar{z}} \quad (\text{III.41})$$

<sup>1</sup>Ce choix nous place dans la lignée des travaux décrits dans [Boyd et al. 1994]. Dans ces travaux, il est établi que l'approximation intérieure d'une région convexe de l'espace par une ellipsoïde est unique et qu'une méthode d'identification de cette ellipsoïde est donnée pour ce cas.

L'obtention d'une représentation par demi-espace d'un tel polytope est un problème de géométrie algorithmique. Il consiste à rechercher l'enveloppe convexe englobant les sommets. De nombreuses publications existent sur le sujet (voir par exemple cf. [Boissonnat & Yvinec 1995] et les références incluses)<sup>2</sup> On obtient alors la représentation du polytope  $\mathcal{P}_n$  suivante :

$$\mathcal{P}_n = \{x \in \mathbb{R}^n \mid P_x x \leq P_c\} \quad (\text{III.42})$$

où  $P_x \in \mathbb{R}^{n_H \times n_z}$  et  $P_c \in \mathbb{R}^{n_H}$  avec  $n_H$  le nombre de demi-espaces composant le polytope. La contrainte d'appartenance de la trajectoire au polytope  $\mathcal{P}_{n_z}$  s'écrit alors :

$$\bar{z}(C, \tau_i) \in \mathcal{P}_{n_z} \Leftrightarrow P_x \bar{z} \leq P_c, \quad i = 1, 2, \dots, \text{card}\{\tau_i\} \quad (\text{III.43})$$

Nous venons de développer dans cette partie, une méthode permettant de remplacer la région admissible  $S_{\bar{z}}$  et les contraintes non linéaires et non convexes associées par des approximations convexes et leurs contraintes convexes associées. Nous entendons tirer un avantage numérique de ces approximations. Cependant il nous paraît nécessaire de pouvoir appréhender la contrepartie de l'utilisation de ces dernières au sein de notre méthodologie de génération de trajectoires.

### III.4.3 Mesure du conservatisme

Les techniques d'approximation présentées précédemment vont naturellement induire un certain degré de pessimisme une fois introduite dans le processus d'optimisation. Il est donc important de pouvoir chiffrer ce conservatisme. Nous proposons un indicateur basé sur l'estimation de la perte d'espace admissible par comparaison des volumes de  $S_{\bar{z}}$  et de l'approximation convexe intérieure  $S_i$  choisie. Un calcul explicite du volume  $S_{\bar{z}}$  semble difficile à appréhender étant donné la complexité des expressions du système d'inégalité (III.12). L'obtention d'une estimation fine du volume de  $S_{\bar{z}}$  passe par le calcul d'une approximation fine capable de retranscrire la non convexité de  $S_{\bar{z}}$ . Comme nous le verrons, l'utilisation des méthodes ensemblistes pour approcher  $S_{\bar{z}}$  permettra d'obtenir une approximation fiable de  $S_{\bar{z}}$  et conduire à un calcul relativement facile du volume de cette approximation. Nous en déduisons un critère pour évaluer le degré de conservatisme des algorithmes.

La méthode d'approximation que nous proposons se base donc sur l'analyse par intervalles. Les bases de l'analyse par intervalles et de l'inversion ensembliste nécessaires à la compréhension de ce qui suit sont données en annexe. Nous avons par exemple tenté d'obtenir une approximation de  $S_{\bar{z}}$  par inversion ensembliste du système (III.12). Or, ce problème de satisfaction de contraintes implique des expressions rationnelles complexes en  $\bar{z}$ . C'est pour cette raison qu'aucune solution n'a pu être obtenue en un temps raisonnable. Toutefois, nous avons développé une méthode de résolution par étapes du problème de satisfaction de contraintes (III.12).

#### III.4.3.1 Algorithme d'approximation de $S_{\bar{z}}$

La méthode présentée dans ce paragraphe se base sur le fait que  $z$ ,  $\dot{z}$  et  $\ddot{z}$  apparaissent de manière récursive dans les trois inégalités de (III.12). En effet, la contrainte (III.6) ne dépend que de  $z$ , la contrainte (III.7) dépend de  $z$  et  $\dot{z}$ , enfin la contrainte (III.8) dépend de  $z$ ,  $\dot{z}$  et  $\ddot{z}$ . Ainsi, la résolution du problème (III.12) sera divisée en trois étapes :

<sup>2</sup>Plusieurs algorithmes sont disponibles sous MATLAB : voir par exemple la librairie de fonctions POLYTOPE de la toolbox MPT [Kvasnica, Grieder & Baotić 2004].

1. Résolution de  $P_1 = \{z | z_1^2 + z_2^2 + z_3^2 \leq 1\}$
2. Résolution de  $P_2 = \{z, \dot{z} | H_{int}(z, \dot{z}) \in \mathbf{H} \text{ et } z \in P_1\}$
3. Résolution de  $P_3 = \{z, \dot{z}, \ddot{z} | \dot{H}_{int}(z, \dot{z}, \ddot{z}) \in \mathbf{C} \text{ et } (z, \dot{z}) \in P_2\}$

**III.4.3.1.1 Résolution du problème de satisfaction de contraintes  $P_1$**  La première étape de la résolution consiste à paver la sphère unité de dimension 3 définie par :

$$z_1^2 + z_2^2 + z_3^2 \leq 1 \quad (\text{III.44})$$

En effet, nous obtenons  $\mathbf{z}_i$ , une liste de pavés en  $z$ , inclus dans la sphère unité à l'aide du solveur REALPAVER [Granvilliers 2004].

**Remarque III.5** Nous considérons à cette étape uniquement les pavés qui sont strictement inclus dans la sphère unité i.e.  $\underline{\mathcal{S}}$ . Ainsi, nous avons :

$$\bigcup_i (z)\text{-box}_i \subsetneq P_1 \quad (\text{III.45})$$

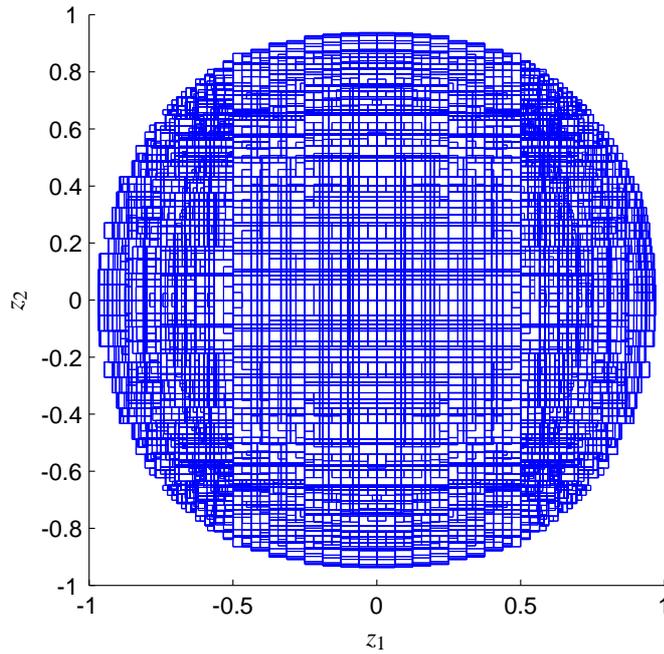


FIG. III.5 – Projection des pavés  $\mathbf{z}_i$  suivant  $z_1$  et  $z_2$

**III.4.3.1.2 Résolution du problème de satisfaction de contraintes  $P_2$**  La seconde étape consiste à résoudre la contrainte (III.7) en  $z$  et  $\dot{z}$  :

$$H_{int}(z, \dot{z}) \in \mathbf{H} \quad (\text{III.46})$$

où  $\mathbb{R}^3 \ni \mathbf{H} = [-H_{max} H_{max}]^3$ . Afin d'obtenir la liste des pavés en  $(z, \dot{z})$  répondant à la contrainte (III.46), nous allons utiliser l'équation de la cinématique (II.56). Il nous faut tout d'abord exprimer

le pavé  $[\Omega]$ . En effet, l'équation (III.46) peut se réécrire comme une contrainte sur  $\Omega$  telle que  $\Omega \in [\Omega]$  puisque sont connus dans l'équation (I.9) la matrice d'inertie  $\mathbb{I}_{PF}$ , le vecteur  $H_{tot}$  et le vecteur d'intervalles  $\mathbf{H}$  :

$$[\Omega] = -\mathbb{I}_{PF}^{-1}(\mathbf{H} - H_{tot}) \quad (\text{III.47})$$

Ensuite, l'équation de la cinématique (II.56) permet d'obtenir la liste des pavés  $\mathbf{z}, \dot{\mathbf{z}}_i$  répondant à la contrainte (III.46) et (III.44) :

$$\dot{\mathbf{z}}_i = \Theta(\mathbf{z}_i)[\Omega], \quad i = 1, \dots, \text{card}\{\mathbf{z}_i\} \quad (\text{III.48})$$

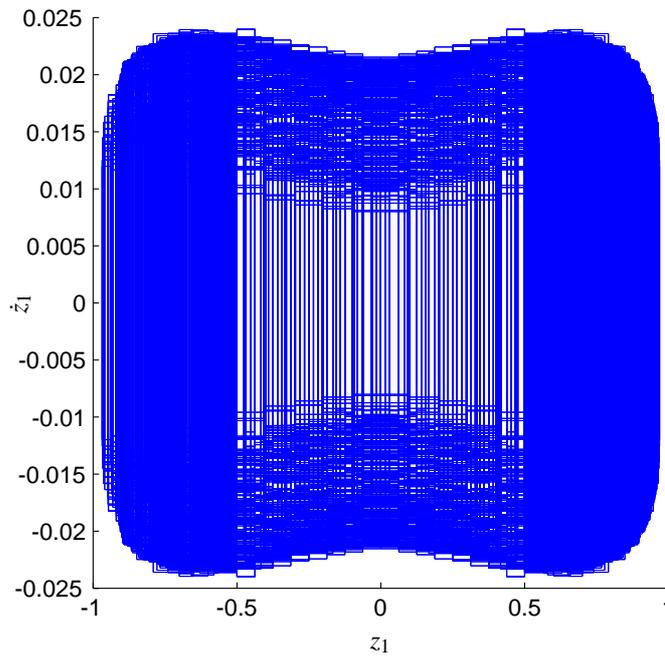


FIG. III.6 – Projection des pavés  $\mathbf{z}, \dot{\mathbf{z}}_i$  suivant  $z_1$  et  $\dot{z}_1$

**III.4.3.1.3 Résolution du problème de satisfaction de contraintes  $P_3$**  La dernière étape consiste à résoudre de la contrainte (III.7) en  $z, \dot{z}$  et  $\ddot{z}$  :

$$\dot{H}_{int}(z, \dot{z}, \ddot{z}) \in \mathbf{C} \quad (\text{III.49})$$

où  $\mathbb{R}^3 \ni \mathbf{C} = [-C_{max} C_{max}]^3$ . En développant l'expression du vecteur  $\dot{H}_{int}(z, \dot{z}, \ddot{z})$ , il apparaît que le couple  $\dot{H}_{int}$  dépend linéairement de  $\ddot{z}$  :

$$\dot{H}_{int}(z, \dot{z}, \ddot{z}) = \alpha(z, \dot{z})\ddot{z} + \beta(z, \dot{z}) \quad (\text{III.50})$$

où  $\alpha \in \mathbb{R}^{(3 \times 3)}$  et  $\beta \in \mathbb{R}^3$ . En associant (III.49) et (III.50), on peut écrire à partir de la liste des pavés en  $(z, \dot{z})$  solutions de  $P_2$  et les pavés en  $(z, \dot{z}, \ddot{z})$  solutions de  $P_3$  :

$$\alpha(\mathbf{z}_i, \dot{\mathbf{z}}_i)\ddot{\mathbf{z}}_i = (\mathbf{C} - \beta(\mathbf{z}_i, \dot{\mathbf{z}}_i)), \quad i = 1, \dots, \text{card}\{\mathbf{z}_i\} \quad (\text{III.51})$$

La liste des pavés en  $\{\mathbf{z}, \dot{\mathbf{z}}, \ddot{\mathbf{z}}\}_i$  formant l'approximation extérieure  $S_O$  est obtenue par l'inversion du système linéaire (III.51). Cette inversion peut être réalisée par différentes méthodes qui

sont disponibles par exemple sous MATLAB dans la toolbox INTLAB [Hargreaves 2002]. On peut notamment citer les méthodes de Krawczyk [R. 1969] de Bliiek [Bliiek 1992] et ses variantes [Rohn 1993],[Ning & Kearfott 1997] et [Neumaier 1999]. Pour diminuer le pessimisme du résultat, il peut être intéressant de croiser les résultats des différentes méthodes d'inversion et les résultats qui peuvent être obtenus en utilisant REALPAVER [Granvilliers 2004].

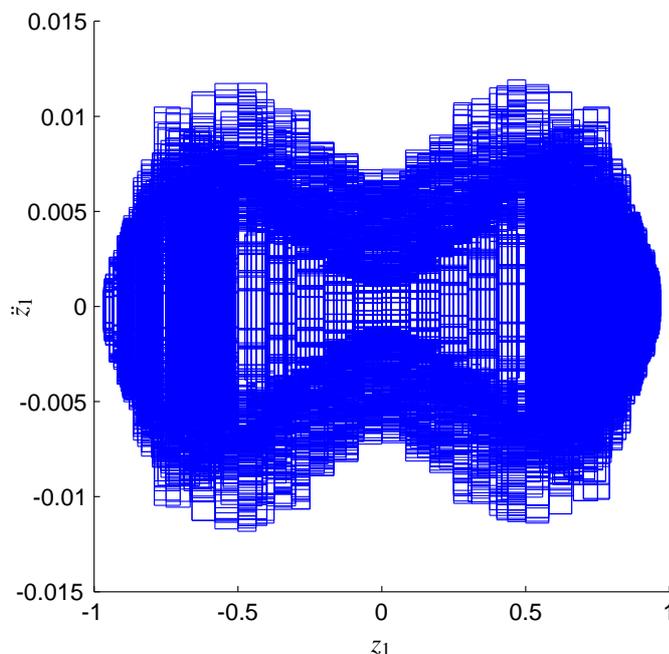


FIG. III.7 – Projection des pavés  $\mathbf{z}, \dot{\mathbf{z}}, \ddot{\mathbf{z}}_i$  suivant  $z_1$  et  $\ddot{z}_1$

**III.4.3.1.4 Volume de  $S_O$**  Le volume de  $S_O$  est le volume de l'union des pavés en  $(z, \dot{z}, \ddot{z})$  formant  $S_O$  :

$$\text{vol}(S_O) = \text{vol} \left( \bigcup_i (z, \dot{z}, \ddot{z})\text{-box}_i \right) \quad (\text{III.52})$$

Rappelons que le volume d'un pavé  $\mathbf{x}$  de dimension  $n$  s'écrit  $\prod_1^{n_z} 2 * \text{rad}(x_i)$  (cf. annexe E). Le volume de  $S_O$  est calculé récursivement en utilisant la loi associative de l'opérateur d'union qui est pour deux ensembles  $A$  et  $B$  :

$$\text{vol}(A \cup B) = \text{vol}(A) + \text{vol}(B) - \text{vol}(A \cap B)$$

La précédente équation est simplifiée puisque

$$(z, \dot{z}, \ddot{z})\text{-box}_i \cap (z, \dot{z}, \ddot{z})\text{-box}_j = \emptyset \text{ avec } i \neq j$$

par le fait que tous les pavés solutions de  $P_1$ , obtenus par REALPAVER ne se recouvrent pas.

Nous avons proposé deux types d'approximations de  $S_{\ddot{z}}$  ce qui nous permettent à présent de mesurer le conservatisme engendré par l'utilisation des algorithmes de génération de trajectoires sous-optimales. On utilisera  $\Delta V$  pour mesurer la perte de volume admissible induite à

l'approximation convexe utilisée.

$$\Delta_V = \left( \frac{\text{vol}(S_O)}{\text{vol}(S_I)} \right)^{\left(\frac{1}{n_{\bar{z}}}\right)} \quad (\text{III.53})$$

L'exposant  $\frac{1}{n_{\bar{z}}}$  permet de ramener le rapport de ces volumes de dimension  $n_{\bar{z}}$  à un rapport unidimensionnel.

## III.5 Etude d'une trajectoire

Cette section est dédiée à l'étude d'une trajectoire d'attitude en temps fixé. L'objectif est de mettre en évidence les différents éléments techniques abordés dans les paragraphes précédents ainsi que les difficultés lors de l'application de la démarche. Nous ferons ensuite une étude numérique approfondie des différentes techniques proposées.

### III.5.1 Définition de la mission

La mission considérée consiste en un ralliement en temps fixé de l'attitude ( $\psi_1 = \psi_2 = \psi_3 = 0^\circ$ ) à l'attitude ( $\psi_1 = \psi_2 = \psi_3 = 30^\circ$ ) en angles de Cardan <sup>3</sup> [Dulot 2003] dans un temps de manoeuvre fixé  $[t_0, t_f]$ . Les points de départ et d'arrivée seront des points d'équilibre du système :

$$\begin{cases} \Omega(t_0) = 0 & \Omega(t_f) = 0 \\ \dot{H}_{int}(t_0) = 0 & \dot{H}_{int}(t_f) = 0 \end{cases} \quad (III.56)$$

La propriété de conservation des points d'équilibre entre deux systèmes Lie-Bäcklund équivalents (cf. théorème II.2 du chapitre II) permet de déduire aisément les conditions initiales et finales du problème de commande optimale (III.10) :

$$\begin{cases} z(t_0) = z_0 & z(t_f) = z_f \\ z^{(s)}(t_0) = z^{(s)}(t_f) = 0, & s \geq 1 \end{cases} \quad (III.57)$$

Le modèle dynamique du satellite rigide est donné par (II.65). La matrice d'inertie  $\mathbb{I}_{PF}$  et le moment cinétique total du système  $H_{tot}$  sont les paramètres de ce modèle. La matrice d'inertie, qui est non diagonale, est représentative du satellite d'observation de la terre JASON développé par THALES ALENIA SPACE.

$H_{tot}$  est choisi de manière à ce que les roues à réaction soient à l'arrêt en début et fin de manoeuvres. Les contraintes d'actionneurs sont données par :

$$\begin{cases} H_{max} = 10 N.m.s \\ C_{max} = 5 N.m \end{cases} \quad (III.58)$$

4

<sup>3</sup>La relation entre les angles de cardan et les quaternions est donnée par :

$$\begin{cases} q_0 = \cos \frac{\psi_1}{2} \cos \frac{\psi_2}{2} \cos \frac{\psi_3}{2} + \sin \frac{\psi_1}{2} \sin \frac{\psi_2}{2} \sin \frac{\psi_3}{2} \\ q_1 = \cos \frac{\psi_1}{2} \cos \frac{\psi_2}{2} \sin \frac{\psi_3}{2} - \sin \frac{\psi_1}{2} \sin \frac{\psi_2}{2} \cos \frac{\psi_3}{2} \\ q_2 = \sin \frac{\psi_1}{2} \sin \frac{\psi_2}{2} \cos \frac{\psi_3}{2} + \sin \frac{\psi_1}{2} \cos \frac{\psi_2}{2} \sin \frac{\psi_3}{2} \\ q_3 = \sin \frac{\psi_1}{2} \cos \frac{\psi_2}{2} \cos \frac{\psi_3}{2} - \cos \frac{\psi_1}{2} \sin \frac{\psi_2}{2} \sin \frac{\psi_3}{2} \end{cases}$$

La relation réciproque est :

$$\begin{cases} \psi_1 = \arctan \frac{2(q_2 q_3 + q_0 q_1)}{2(q_0^2 + q_1^2) - 1} \\ \psi_2 = -\arcsin(2q_1 q_3 - 2q_0 q_2) \\ \psi_3 = -\arctan \frac{2(q_2 q_3 + q_0 q_1)}{2(q_0^2 + q_3^2) - 1} \end{cases} \quad (III.55)$$

Les angles  $\psi_1$  et  $\psi_3$  sont indéterminés si  $\cos \psi_2 = 0$

<sup>4</sup>L'ensemble des résultats présenté ci-dessous a été obtenu sur un ordinateur muni d'un processeur Intel P4

### III.5.2 Etude des paramètres de configuration

Les paramètres d'entrée des différents algorithmes de génération de trajectoires comportent, hormis les données de la mission, les paramètres de définition des B-splines et la séquence de points de collocation.

#### III.5.2.1 Paramètres de définition des B-splines

Afin de définir une base de B-splines, il est nécessaire de choisir l'ordre  $k$  et de définir une séquence nodale  $T$  basée sur une séquence de points distincts  $\xi$  divisant l'intervalle de temps  $[t_0, t_f]$  en  $l$  segments :

$$\begin{aligned} T &= \{t_1, \dots, t_L\} \\ &= \underbrace{\{\xi_1, \dots, \xi_1\}}_{\rho_1 \text{ fois}}, \underbrace{\{\xi_2, \dots, \xi_2\}}_{\rho_2 \text{ fois}}, \dots, \underbrace{\{\xi_{l+1}, \dots, \xi_{l+1}\}}_{\rho_{l+1} \text{ fois}} \end{aligned} \quad (\text{III.59})$$

Nous allons étudier tout d'abord l'influence du choix de la séquence nodale  $T$  sur la trajectoire. Sans connaissance *a priori* sur la trajectoire, le plus simple est de distribuer uniformément les  $l$  segments temporels. Ensuite, nous désirons que les trajectoires des sorties plates soient de classe de continuité maximale sur l'intervalle ouvert  $]t_0, t_f[$  et qu'elles soient discontinues en  $t_0$  et  $t_f$ . Or la relation entre ordre  $k$ , multiplicité  $\rho_i$  et continuité définie dans l'annexe C nous impose alors les multiplicités suivantes :

$$\rho_i = \begin{cases} k, & \text{pour } i = 1, l+1 \\ 1, & \text{pour } i = 2, \dots, l \end{cases} \quad (\text{III.60})$$

Le nombre de coefficients est donné par la relation  $n = k.l - \sum_{i=1}^{l+1} \nu_i$  (cf. annexe C). Prenons pour exemple une base de B-splines d'ordre 6 et définie sur 5 segments. En considérant les multiplicités (III.60), nous obtenons une base de 10 B-splines. Nous choisissons les points de collocation tels que les contraintes soient vérifiées à une fréquence d'un point par seconde. La résolution du problème d'optimisation (III.18) permet d'obtenir les profils temporels de la figure III.8. Il est intéressant de noter que la trajectoire se déroule en trois phases. On distingue clairement deux phases d'accélération séparées par une phase de croisière. Cette phase de croisière dure environ 8 secondes. Durant les phases d'accélération, les profils de couples n'utilisent que 40 à 60% des couples alloués par les axes.

Il peut être intéressant de réduire les phases d'accélération pour permettre d'allonger la phase de croisière. Pour ce faire, nous allons modifier la séquence nodale. Nous conservons 5 segments, mais nous décidons d'agrandir significativement le segment central par la distribution de la séquence de segmentation  $\xi$  suivante :

$$\xi = \{0, 2, 4, 21, 23, 25\}$$

En conservant les autres paramètres d'entrées, l'algorithme d'optimisation fournit les profils de la figure III.9. Par ce placement particulier des noeuds de segmentation, nous allongeons significativement la phase de croisière pour utiliser au mieux les couples moteurs sur des phases d'accélération plus courtes.

---

2GHz et de 1Go de mémoire vive, sous un système d'exploitation Debian Linux. La plateforme de génération de trajectoires a été programmée pour le logiciel Matlab et le moteur d'optimisation est fourni par l'Optimization toolbox (l'utilisation des algorithmes SNOpt est envisagée pour les prochaines versions de la plateforme).

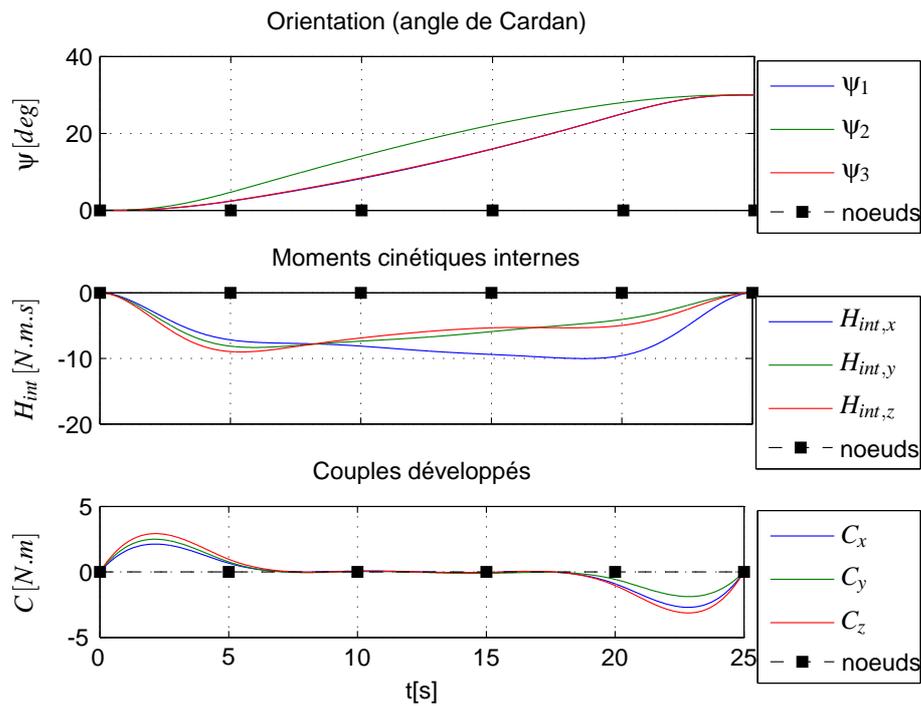


FIG. III.8 – Profils de trajectoire pour une base de B-splines d'ordre 6

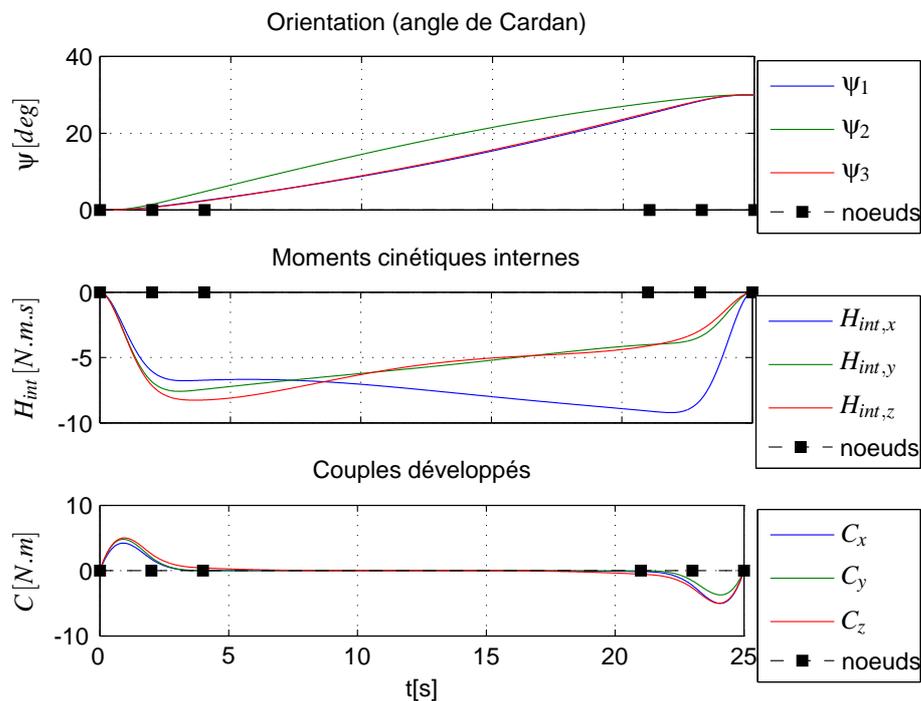


FIG. III.9 – Profils de trajectoire pour une base de B-splines d'ordre 6

### III.5.2.2 Distribution des points de collocation

La détermination des points de collocation est importante pour plusieurs raisons. D'une part, c'est aux points de collocation que sont vérifiées les contraintes (III.12) et, d'autre part, leur nombre définit la taille du problème d'optimisation avec la dimension du vecteur  $C$  (III.18).

Rappelons que, lors du processus d'optimisation, les contraintes sont considérées comme satisfaites entre deux points de collocation où elles sont vérifiées. Cet état de fait peut déboucher sur des erreurs qui ne peuvent être détectées que *a posteriori*. Prenons l'exemple du ralliement défini précédemment, devant être effectué en 20 secondes à partir d'une base de B-splines d'ordre 6 sur 5 segments uniformes et 20 points de collocations. L'algorithme d'optimisation nous assure du ralliement de l'état final et de la satisfaction des contraintes sur la trajectoire et nous propose les profils de la figure III.10.

On peut remarquer en particulier qu'un phénomène important se produit : les couples enfreignent les contraintes dans les phases d'accélération entre deux points de collocation. Ce constat est à l'origine des développements présentés au chapitre suivant.

Pour pallier ce problème, on peut augmenter uniformément le nombre de points dans le maillage de collocation, mais ceci se fera au détriment de la taille du problème qui va croître. La solution sera de distribuer les points de collocation suivant la contrainte à respecter. En effet, les couples doivent d'être vérifiés lors des phases d'accélération et les moments cinétiques lors de la phase de croisière. On peut ainsi conserver la dimension du problème d'optimisation tout en resserrant le maillage aux endroits où cela est le plus utile (cf figure III.11).

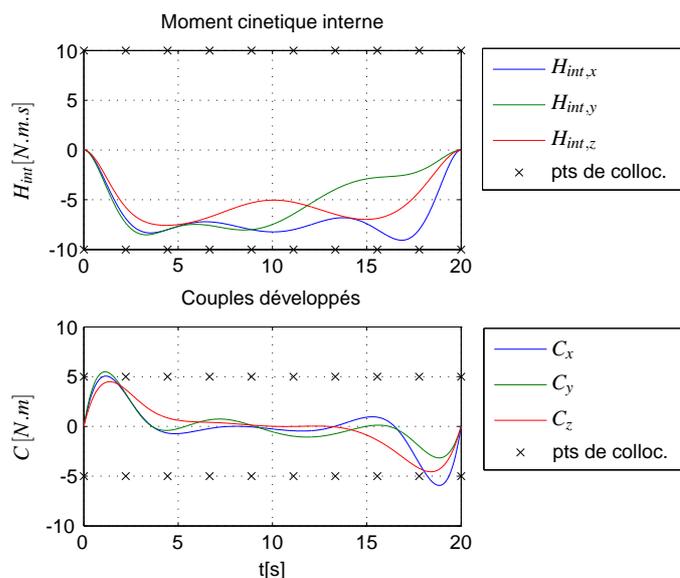


FIG. III.10 – Profils de trajectoire pour une base de B-splines d'ordre 7 et 10 points de collocation distribués uniformément

### III.5.3 Optimisation non convexe

Nous étudierons dans ce paragraphe la procédure de génération de trajectoires basée sur la résolution du problème (III.18) d'un point de vue numérique. L'une des principales préoccupations de la problématique du guidage est d'obtenir un temps minimal de transfert entre deux attitudes. Nous allons mener cette étude pour un ensemble de configurations du problème d'optimisation. Nous nous focaliserons uniquement sur l'influence des paramètres de définition des

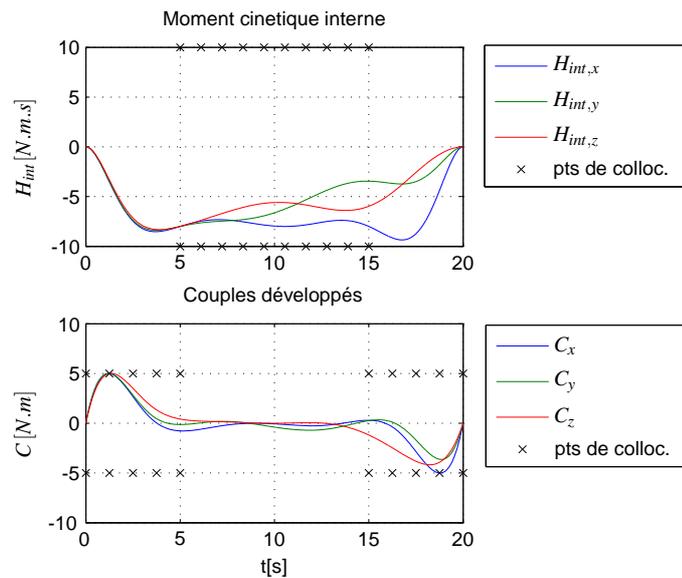


FIG. III.11 – Profils de trajectoire pour une base de B-splines d'ordre 6 et 10 points de collocation avec une distribution adaptée

B-splines, à savoir l'ordre et le nombre de segments (les segments seront distribués uniformément) : l'ordre sera compris entre 4 et 12 et le nombre de segments entre 5 et 10. Nous fixons le nombre de points de collocation à 50. Le processus d'optimisation du temps de transfert est itératif.

A chaque itération, le problème d'optimisation (III.18) est résolu. Si une solution admissible est trouvée, alors le temps de transfert est réduit. Si aucune solution n'est trouvée, alors on considérera que le dernier temps de transfert ayant obtenu une solution est le temps minimal de manoeuvre. Les résultats sont présentés sur le graphique (III.13). Les temps de transfert s'échelonnent entre 27.8 pour un ordre 4 sur 5 segments à 18.7 secondes pour un ordre 12 sur 10 segments. On observe sur le graphe (III.13) que l'ordre des B-splines est un facteur important dans la réduction du temps de transfert. On peut aussi noter que le nombre croissant de segments augmente la performance de l'algorithme d'optimisation. Ces résultats peuvent notamment s'expliquer par l'augmentation de la dimension de la base des B-splines.

D'un point de vue numérique, les données que nous allons observer sont le nombre d'itérations et le temps nécessaire à l'algorithme de programmation non linéaire pour converger vers un résultat acceptable. Pour ce faire, nous allons résoudre le problème d'optimisation (III.18) pour un temps de transfert de 30 secondes. Nous allons étudier l'influence des paramètres de configuration de la procédure de génération de trajectoires sur le nombre d'itérations et le temps.

Le graphique (III.14) permet d'observer le nombre d'itérations en fonction de l'ordre des B-splines, du nombre de points de collocation et du nombre de segments sur la trajectoire. Dans le cas où l'intervalle de temps est divisée en 5 segments, il apparaît que l'ordre des B-splines est le paramètre le plus influent : le nombre d'itérations passe d'une dizaine à plus de 90 lorsque l'ordre va de 4 à 12 et quelque soit le nombre de points de collocation utilisés dans le processus. Il semble que l'accroissement du nombre de points de collocation n'apporte pas de complexité supplémentaire au problème.

Nous pouvons faire un constat équivalent lorsque 10 segments sont utilisés. L'ordre des B-splines s'avère le facteur déterminant dans l'évolution du temps de calcul. Ceci dit, on peut

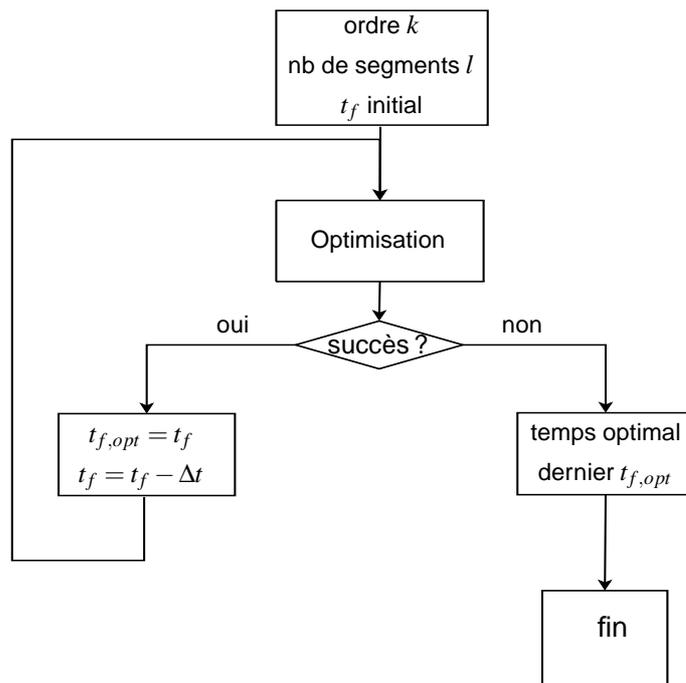


FIG. III.12 – Processus de recherche du temps optimal

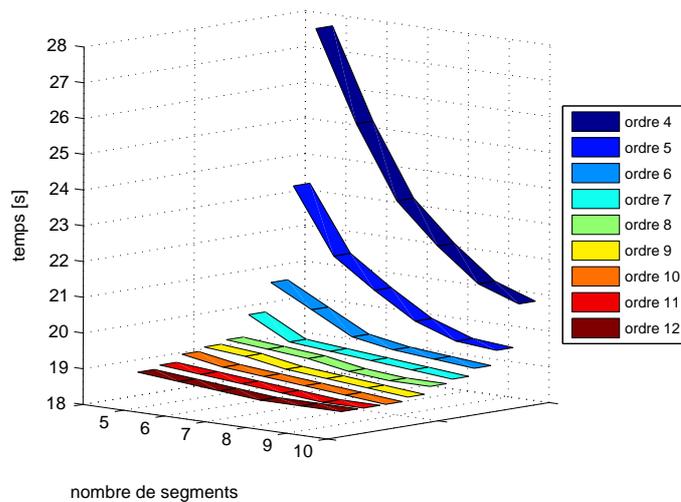


FIG. III.13 – Temps optimal de transfert

remarquer que l'accroissement du nombre de points de collocation influence également le temps de calcul. Cela peut s'expliquer par le fait qu'à nombre égal d'itérations, le nombre d'opérations élémentaires effectuées est fonction de la dimension du vecteur de collocation.

Ainsi, c'est principalement la dimension de la base de B-splines qui détermine intrinsèquement la complexité du problème d'optimisation à résoudre. L'influence du nombre de points de collocation est visible quand on s'attache au temps de calcul. En termes d'itérations effectués, elle n'apparaît que dans le cas d'un nombre important de variables à optimiser. Ce phénomène peut s'expliquer par les tailles importantes des matrices jacobiniennes utilisées dans les algorithmes de programmation non linéaire, ce qui nécessite un certain nombre de traitements numériques (méthodes de réductions de matrices creuses, cf annexe A).

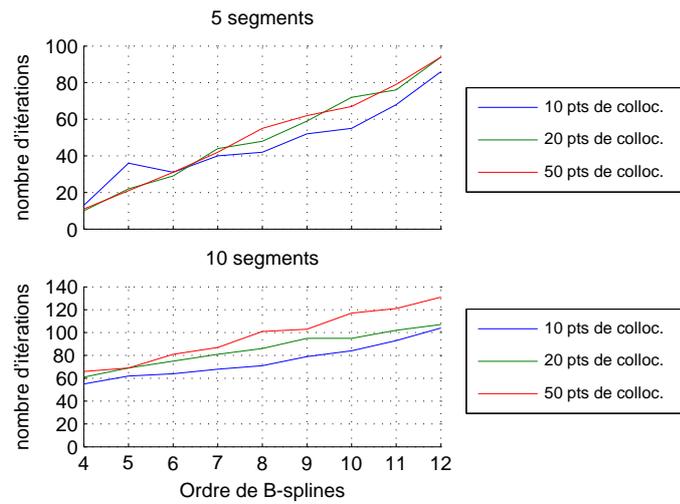


FIG. III.14 – Nombre d'itérations pour la procédure non convexe

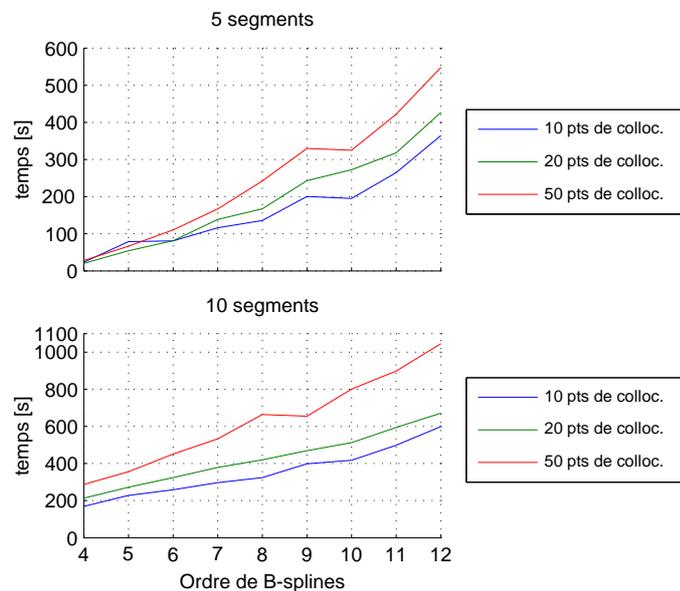


FIG. III.15 – Temps d'exécution pour la procédure non convexe

### III.5.4 Optimisation convexe

Nous avons développé, au cours de la section III.4, plusieurs méthodes d'approximation de l'espace admissible. Cela nous a permis de mettre en place une procédure de génération de trajectoires, basée sur le problème d'optimisation (III.34). Nous allons étudier le comportement numérique de cette procédure pour différents cas d'approximation convexe.

#### III.5.4.1 Contraintes quadratiques

Comme nous l'avons déjà précisé, les ellipsoïdes sont un cas particulier des superellipsoïdes dans le sens où elles sont les seules superellipsoïdes qui permettent de décrire leur fonction d'appartenance comme une contrainte matricielle quadratique (cf (III.39)).

Nous allons tout d'abord étudier l'influence d'une telle approximation sur le temps optimal de

transfert. L'étude est semblable à celle menée dans le paragraphe III.5.3. Les résultats obtenus sont représentés sur le graphique III.16.

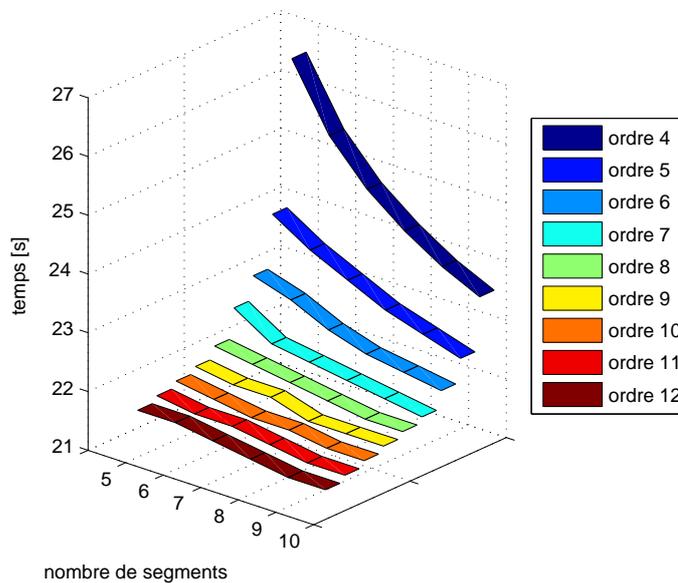


FIG. III.16 – Temps optimal de transfert pour une approximation ellipsoïdale

Le temps minimal obtenu est de 21.5 secondes pour l'ordre 12 ce qui est supérieur d'environ 3 secondes au temps optimal dans le cas non convexe présenté sur la figure III.13. Ceci s'explique par l'utilisation de l'approximation ellipsoïdale dans le problème (III.34). Les tendances observées du temps minimal de transfert vis-à-vis de l'ordre des B-splines et du nombre de segments sont sensiblement les mêmes que les résultats obtenus avec la procédure non convexe III.18.

L'apport numérique de la méthode est observable sur les graphiques III.17 et III.18. Le nombre d'itérations est maintenu sous le seuil des 15 itérations pour 5 segments. Pour 10 segments, la majorité des optimisations se sont déroulées en moins de 30 itérations. Notons que ces résultats sont meilleurs que ceux de la procédure non convexe en termes du nombre d'itérations. Le même constat peut être fait pour le temps d'exécution. Il est maintenu sous les 50 secondes pour 5 segments et sous les 100 secondes pour une majorité des optimisations produites. D'un point de vue qualitatif, les profils de trajectoires observés sur la figure III.19 ont des comportements qualitatifs équivalents à ceux obtenus dans la figure III.8. Cette observation nous permet de justifier *a posteriori* l'évaluation de la fonction de coût (III.37).

#### III.5.4.2 Contraintes linéaires

Dans ce paragraphe, nous étudierons le problème sous optimal (III.34), soumis à des contraintes linéaires, dans lequel la surface d'approximation est donnée par  $\mathcal{S}_6$  (cf. paragraphe III.4.2.3). L'étude du temps minimal de transfert est réalisée avec des contraintes linéaires. Les résultats sont exposés sur le graphique (III.20). L'utilisation de  $\mathcal{S}_6$  pénalise fortement le temps de transfert. En effet, le temps minimal obtenu (36 secondes) est le double de ce qui a pu être obtenu dans le paragraphe III.5.3. On retrouve également l'influence de l'ordre et du nombre des segments de trajectoires sur les résultats.

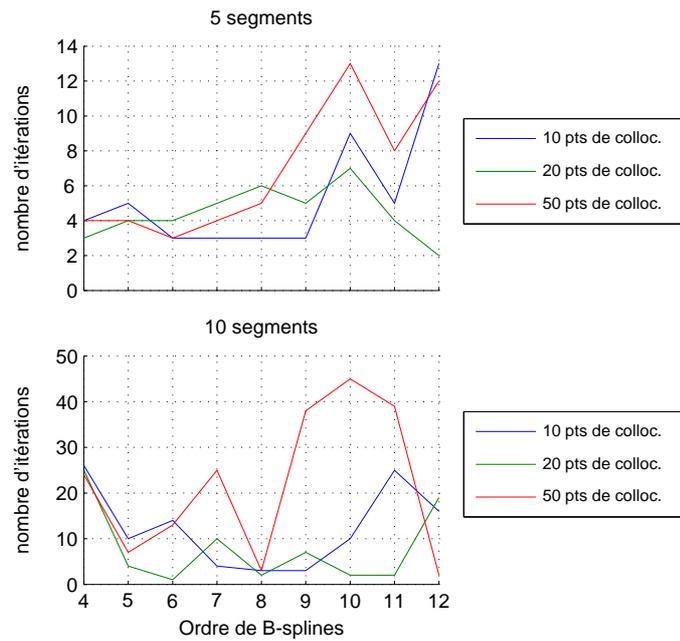


FIG. III.17 – Nombre d'itérations sous contraintes quadratiques

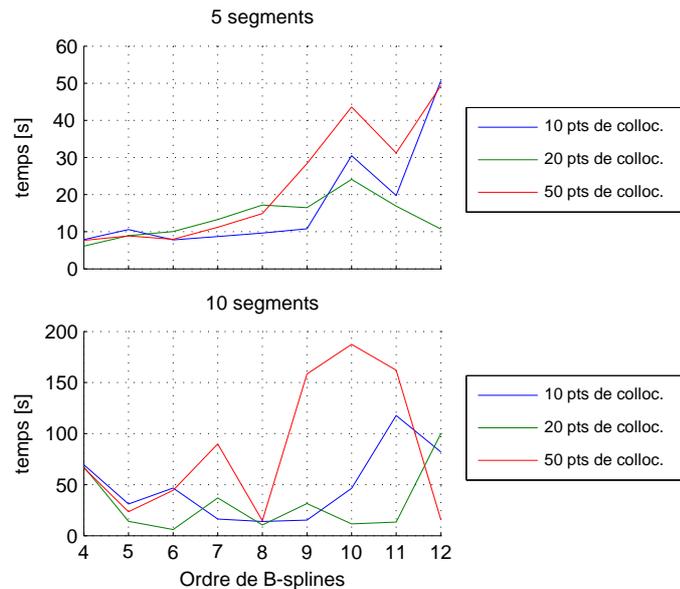


FIG. III.18 – Temps d'exécution sous contraintes quadratiques

Le principal intérêt de l'utilisation de ce polytope est numérique. Nous avons relevé le nombre d'itérations et le temps nécessaire à l'optimisation pour un temps de transfert de 50 secondes. Les résultats sont données sur les figures III.21 et III.22. Le nombre d'itérations est limité à 2 et le temps d'exécution à 20 secondes.

### III.5.4.3 Contraintes non linéaires

Nous allons étudier dans ce paragraphe l'apport de l'utilisation des superellipsoïdes dans le problème d'optimisation (III.34). Pour ce faire, nous allons nous concentrer sur un exemple de superellipsoïdes issues de l'étude du tableau III.1. La superellipsoïde étudiée est la surface  $S_1$

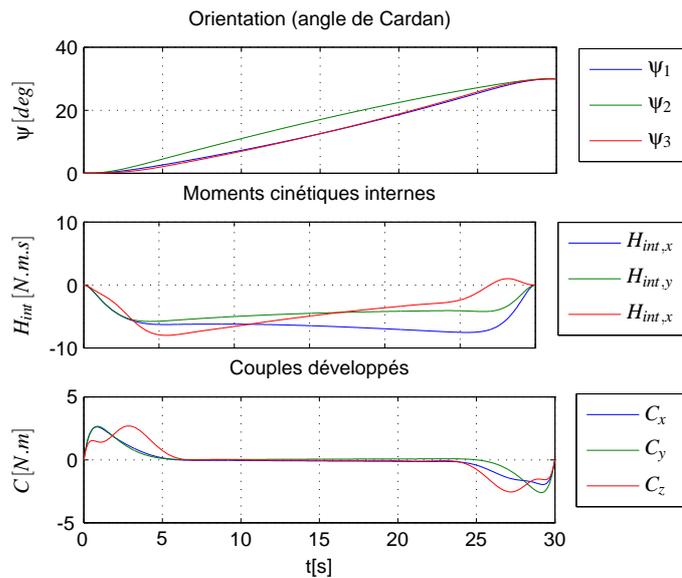


FIG. III.19 – Profils de trajectoire obtenus sous contraintes quadratiques

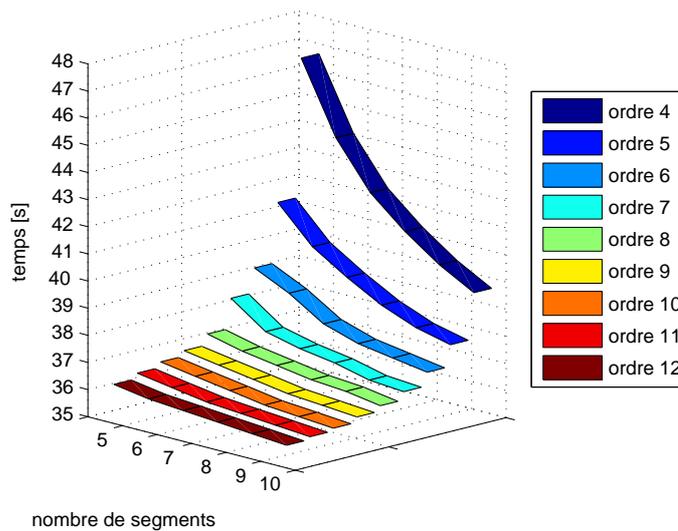


FIG. III.20 – Temps minimal de transfert sous contraintes linéaires

dont le volume intérieure est le plus important obtenu dans l'étude du tableau III.1.

Les résultats de l'étude de temps minimal de transfert sont donnés sur le graphique III.23. Le temps minimal obtenu est de 19.6 secondes, ce qui est environ une seconde plus long que les résultats obtenus avec la procédure (III.18). On peut aussi noter que les performances en termes de temps de transfert sont meilleures que pour une approximation ellipsoïdale. Ce fait peut s'expliquer par le volume plus important de la forme  $S_1$  par rapport à l'ellipsoïde  $S_4$ . Comme pour les études précédentes, l'ordre de la base des B-splines est le facteur prédominant dans la réalisation d'un temps de transfert optimal.

D'un point de vue numérique, les résultats sont donnés sur les graphiques III.24 et III.25. On peut observer que le nombre d'itérations est faible : il faut moins d'une dizaine d'itérations pour obtenir un résultat avec 5 segments et moins d'une trentaine avec 10 segments. Les temps d'exécution sont corrélés et il faut moins de 80 secondes pour terminer l'optimisation avec succès pour 10

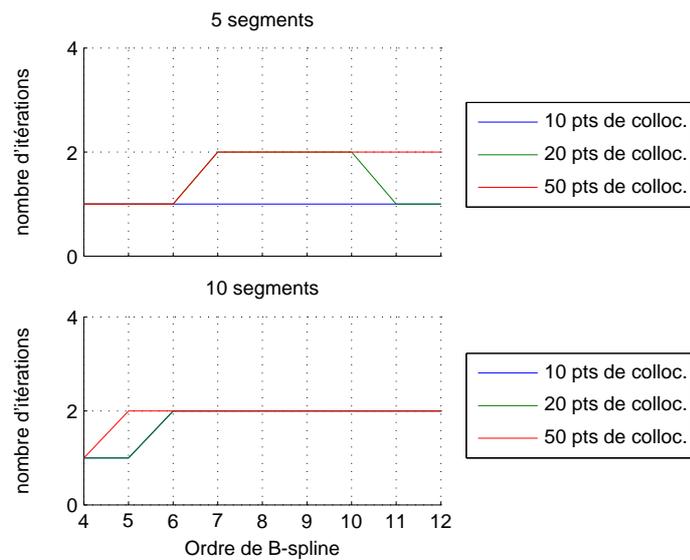


FIG. III.21 – Nombres d'itérations sous contraintes linéaires

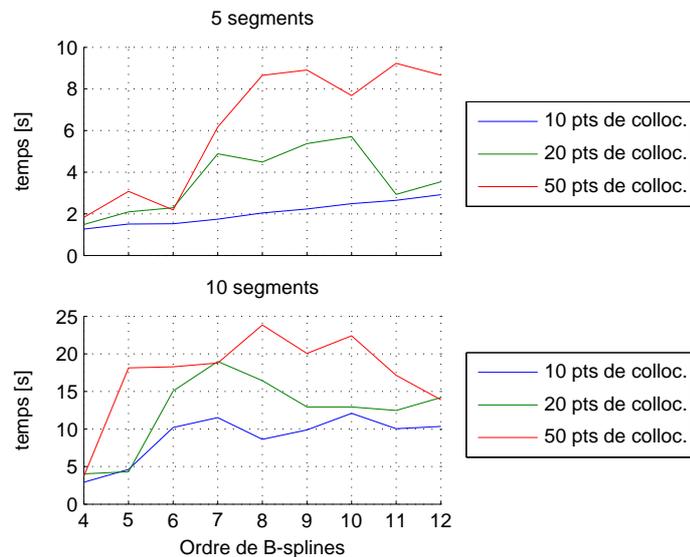


FIG. III.22 – Temps d'exécution sous contraintes linéaires

segments et moins de 50 secondes avec 5 segments.

Pour finir, nous avons étudié l'influence des superellipsoïdes  $S_2$ ,  $S_4$  et  $S_5$  sur le temps optimal de manoeuvre. Les résultats sont exposés sur les figures III.26 III.27 III.28. On constate la dégradation du temps optimal de transfert avec la diminution du volume d'approximation. De manière plus générale, on peut établir un fort lien entre le volume de l'approximation et les performances en temps de transfert.

### III.5.5 Comparaison des différents algorithmes

Nous avons présenté les résultats de trois algorithmes sous optimaux utilisant des approximations superellipsoïdales. Deux de ces approximations présentent la particularité de définir des contraintes d'appartenance à la région admissible quadratique et linéaire.

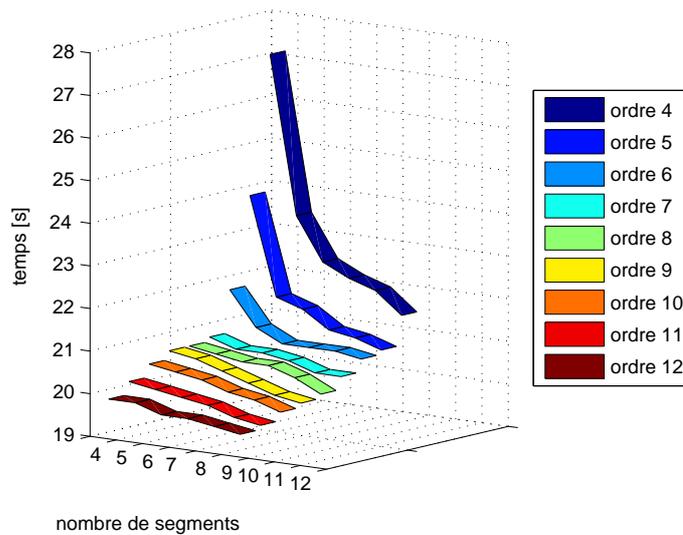


FIG. III.23 – Temps minimal de transfert sous approximation superellipsoïdale  $S_1$

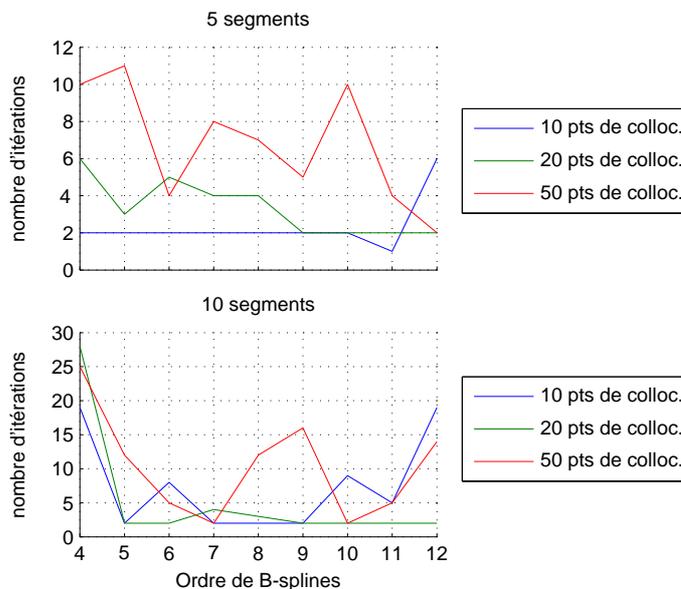


FIG. III.24 – Nombre d'itérations sous contraintes superellipsoïdales

D'un point de vue optimisation du temps de transfert, nous pouvons faire le constat que le temps minimal obtenu est lié au volume de l'approximation : plus le volume est important et se rapproche du volume de  $S_{\bar{z}}$ , plus la manoeuvre se fait en un temps minimal.

Concernant la charge de calcul, notre analyse porte sur la comparaison des indicateurs du tableau III.2  $\Delta V$  et  $\Delta_{CPU}$ .  $\Delta_{CPU}$  représente le rapport entre les temps d'exécution des algorithmes de génération de trajectoires optimale et sous optimale.

$$\Delta_{CPU} = \frac{\text{temps CPU d'optimisation globale}}{\text{temps CPU d'optimisation convexe}} \quad (III.61)$$

L'utilisation des contraintes linéaires dans le problème (III.34), permet de limiter de manière importante le nombre d'itérations et le temps d'exécution. Les deux autres algorithmes ont des

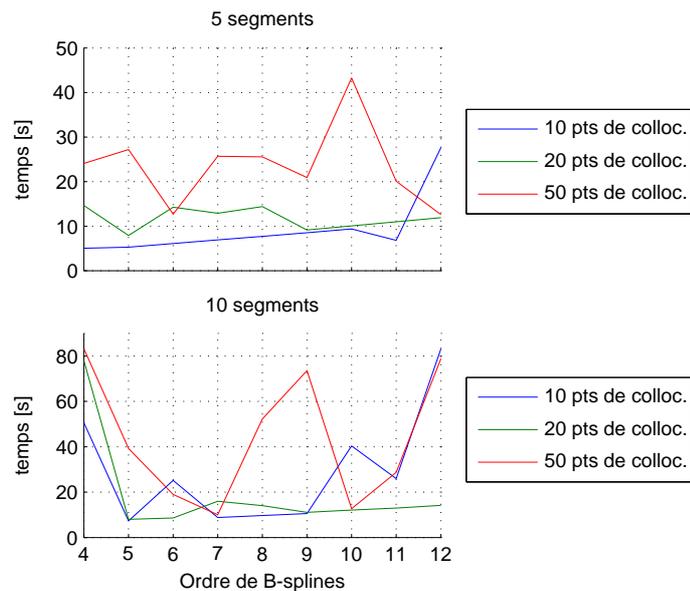
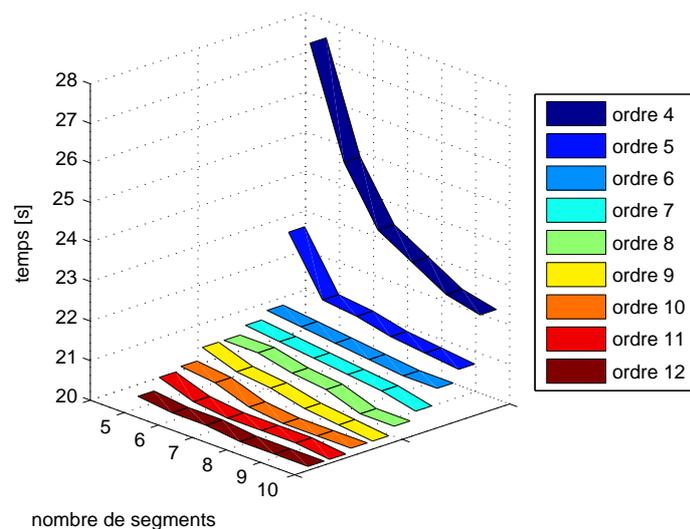
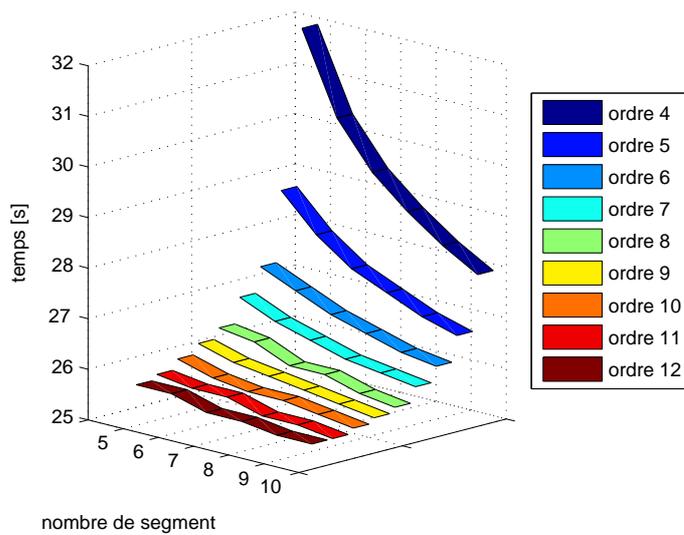
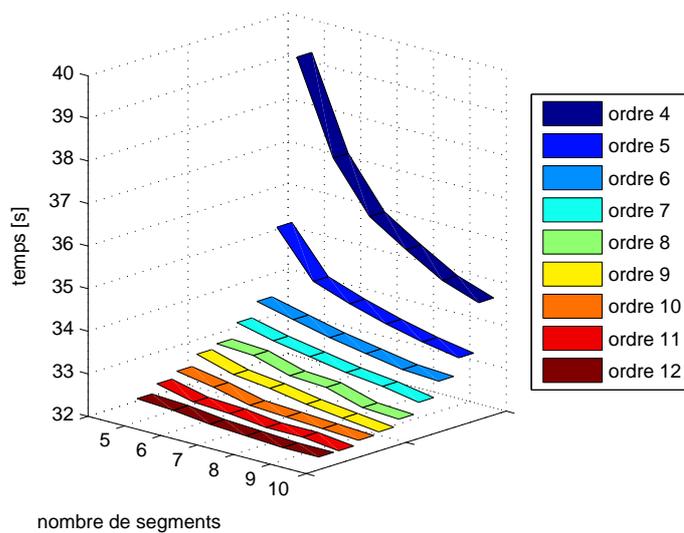


FIG. III.25 – Temps d'exécution sous contraintes superellipsoïdales

FIG. III.26 – Temps minimal de transfert sous approximation superellipsoïdale  $\mathcal{S}_2$ 

temps moyens d'exécution équivalents.

A  $\Delta_{CPU}$  équivalents, il est plus intéressant d'utiliser des contraintes non linéaires ( $\mathcal{S}_1$ ) plutôt que quadratiques ( $\mathcal{S}_3$ ) car elles permettent de recouvrir une région de solution plus grande. Le choix entre les approximations  $\mathcal{S}_1$  et  $\mathcal{S}_6$  dépendra de l'application. Dans le cadre de la planification de missions où le nombre de manoeuvres à calculer est important et où la faisabilité est le critère déterminant, les contraintes linéaires présentent des avantages numériques indéniables. Au contraire, si l'on désire définir une manoeuvre particulière ou obtenir un temps minimal de transfert, il paraît important de pouvoir balayer la région admissible la plus grande possible tout en maîtrisant le temps de calcul.

FIG. III.27 – Temps minimal de transfert sous approximation superellipsoïdale  $\mathcal{S}_4$ FIG. III.28 – Temps minimal de transfert sous approximation superellipsoïdale  $\mathcal{S}_5$ 

## III.6 Conclusions

Dans ce chapitre, nous avons développé des techniques d'optimisation de trajectoires basées sur la platitude du modèle d'un satellite rigide. L'efficacité de ces outils dépendant à la fois de la taille du problème et de la convexité de la région admissible, nous avons développé une procédure d'approximation convexe de la région admissible aux contraintes. Cette procédure présente deux avantages : d'une part le nombre de paramètres à optimiser est de l'ordre de la dimension de l'espace des sorties plates, et d'autre part, elle est facilement applicable dans les cas où la dimension des sorties plates est importante. Enfin, nous avons mis en place un critère de comparaison des différents algorithmes permettant de chiffrer leur degré de conservatisme.

Nous avons ensuite appliqué les différentes techniques développées au cas du guidage. Une analyse circonstanciée des résultats de simulation a permis de comparer les performances des

Approximation	superellipsoïde $\mathcal{S}_1$	ellipsoïde $\mathcal{S}_3$	polytope $\mathcal{S}_6$	Non convexe
temps moyen de calcul	36.51	31.89	8.43	336.23
$\Delta_{CPU}$	10.54	9.20	39.88	× × ×
$\Delta_V$	1.48	2.37	5.63	× × ×

TAB. III.2 – Comparaison des algorithmes

différents algorithmes. L'ensemble des algorithmes développés dans ce chapitre présentent un inconvénient intrinsèquement lié à la procédure de collocation : les contraintes ne sont pas vérifiées sur le continuum de temps, mais en un nombre fini d'instants, ce qui implique une analyse *a posteriori* systématique des profils obtenus.

Dans le chapitre V, nous proposons une solution qui permettra de s'affranchir de ce problème, en proposant une formulation alternative du problème de commande optimale des systèmes plats. Cette solution est basée sur la transformation du problème de programmation semi-infinie en un problème de programmation semi-définie positive.

En attendant, dans le chapitre suivant, nous mettons en application les méthodes basées sur la platitude et la collocation sur un simulateur du système de contrôle d'attitude de la mission DEMETER.

---

## Chapitre IV

# Mise en oeuvre : application sur simulateur

---

### Sommaire

---

<b>IV.1</b>	<b>Introduction</b>	<b>86</b>
<b>IV.2</b>	<b>Générateur de trajectoires MANIAC</b>	<b>86</b>
<b>IV.3</b>	<b>Simulations</b>	<b>87</b>
IV.3.1	Description du simulateur	87
IV.3.2	Environnement et conditions de simulations	89
IV.3.3	Solutions calculées par platitude et collocation	89
IV.3.4	Discussion et comparaison des résultats de simulation	89
<b>IV.4</b>	<b>Conclusions</b>	<b>92</b>

---

## IV.1 Introduction

Ce court chapitre a pour objet de présenter les résultats obtenus lors de deux campagnes d'essais en simulation réalisés au CNES. Nous nous étions fixés deux objectifs à atteindre à travers ces essais. Le premier objectif était de valider les profils de guidage auxquels nous étions parvenus grâce aux méthodes développées dans le chapitre III. Le second objectif était d'évaluer, par comparaison avec les outils de génération de trajectoires internes au CNES, l'apport de nos techniques. Pour ce faire, nous avons basé notre étude sur la simulation d'un scénario de manoeuvre d'attitude en temps fixé. Cette opération fut menée à l'aide d'un simulateur dédié et développé au CNES pour le satellite DEMETER. Aussi, après avoir décrit le générateur de trajectoires MANIAC, nous présenterons les résultats de simulation obtenus.

## IV.2 Générateur de trajectoires MANIAC

Durant les années 90, le CNES a développé une méthodologie pour répondre au problème de génération de manoeuvres d'attitude dans le cadre d'un système de contrôle d'attitude basé sur l'utilisation des roues à réactions. Cette méthodologie a notamment été adaptée pour la plate-forme PROTEUS<sup>1</sup> et MYRIADE<sup>2</sup>. L'objectif est de rallier une position finale  $Q_f$  à partir d'une position initiale  $Q_i$  en un temps donné tout en respectant les conditions de vitesse initiale et finale,  $\Omega_i$  et  $\Omega_f$ , ainsi que les contraintes liées aux actionneurs. La méthode consiste à décomposer la manoeuvre en trois rotations, chacune remplissant une tâche :

- rallier l'attitude  $Q_i$  à  $Q_f$  sous la contrainte de vitesses de rotation initiale et finale nulles,
- rallier la vitesse finale  $\Omega_f$  à partir d'une vitesse initiale nulle sans mouvement d'attitude,
- rallier une vitesse finale nulle à partir de la vitesse initiale  $\Omega_i$  sans mouvement d'attitude.

Une telle décomposition de la manoeuvre mène à une formulation analytique de chaque rotation élémentaire ainsi que des moments cinétiques et couples associés, en fonction des conditions initiales et finales. Il est ainsi possible de déterminer chaque rotation et de vérifier si les contraintes sont satisfaites. Le temps minimal de manoeuvre est obtenu par le biais d'un processus itératif.

La durée de manoeuvre est divisée en segments durant lesquels une des six contraintes (moments cinétiques ou couples) est saturée. On change de contrainte saturée aux instants de commutation définissant les segments. De plus, par le biais de contraintes supplémentaires, il est possible d'imposer jusqu'à un degré 3 de continuité sur la trajectoire en début et fin de manoeuvre ainsi qu'aux instants de commutation.

L'algorithme délivre les profils temporels du vecteur vitesse,  $\Omega(t)$ . Ces profils sont des polynômes par morceaux sur les segments associés aux instants de commutation. Ces polynômes par morceaux peuvent atteindre un ordre 6. Le générateur MANIAC propose pour Demeter la manoeuvre définie par les profils de la figure IV.1.

---

<sup>1</sup>Plateforme pour minisatellites (de masse inférieure à 500 Kg) adaptable et reconfigurable, conçue pour différentes missions scientifiques telles que JASON 1 (lancé en décembre 2001), CALLIPSO (avril 2006) et COROT

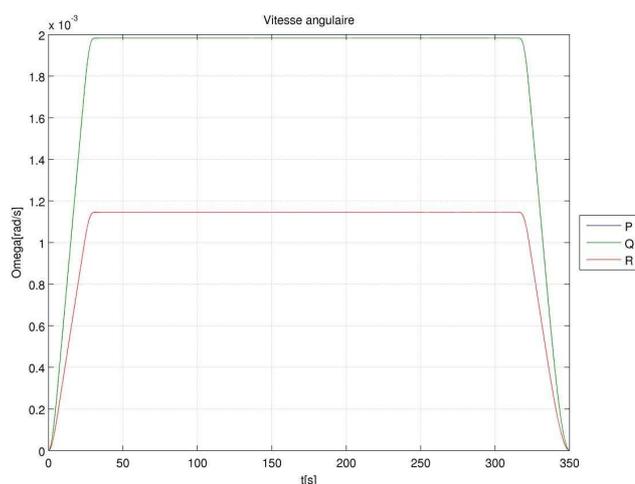


FIG. IV.1 – Profils de vitesse calculés par MANIAC

## IV.3 Simulations

La mission que nous nous proposons d'étudier est identique à celle étudiée dans la section III.5 du chapitre III. Il s'agit d'une manoeuvre d'attitude en temps fixé de l'attitude ( $\psi_1 = \psi_2 = \psi_3 = 0^\circ$ ) à l'attitude ( $\psi_1 = \psi_2 = \psi_3 = 30^\circ$ ) en angles de Cardan dans un temps de manoeuvre fixé  $[t_0, t_f]$ . Les points de départ et d'arrivée seront des points d'équilibre i.e. les vitesses et accélérations sont nulles dans le repère du véhicule.

Le modèle de satellite utilisé représente la dynamique du satellite DEMETER. 40% du moment cinétique et du couple de chacune des roues à réaction est alloué à la manoeuvre. Il vient ainsi :

$$-H_{max} \leq H_{int} \leq H_{max} \quad (IV.1)$$

$$-C_{max} \leq \dot{H}_{int} \leq C_{max} \quad (IV.2)$$

avec  $H_{max} = 0.05 N.m.s$  et  $C_{max} = 0.002 N.m$ .

### IV.3.1 Description du simulateur

Les différents profils de trajectoires sont simulés durant la phase de vol normal. Ainsi le processus va consister à simuler un comportement de l'ensemble des éléments du Système de Contrôle d'Attitude sur Orbite (SCAO) du mode de vol normal (MNO) (voir la figure IV.2).

Afin de situer les simulations dans un contexte réaliste, le calcul des consignes de guidage dans le simulateur présente plusieurs particularités :

- L'orbite est connue à bord grâce à un modèle d'orbite.
- La consigne de guidage du satellite doit être calculée à 4Hz. Elle est représentée par les 2 paramètres suivants :

---

(decembre 2006)

<sup>2</sup>Plateforme modulaire pour microsattelites (de masse inférieure à 150 Kg). Elle a été utilisée pour des missions de type DEMETER et PARASOL

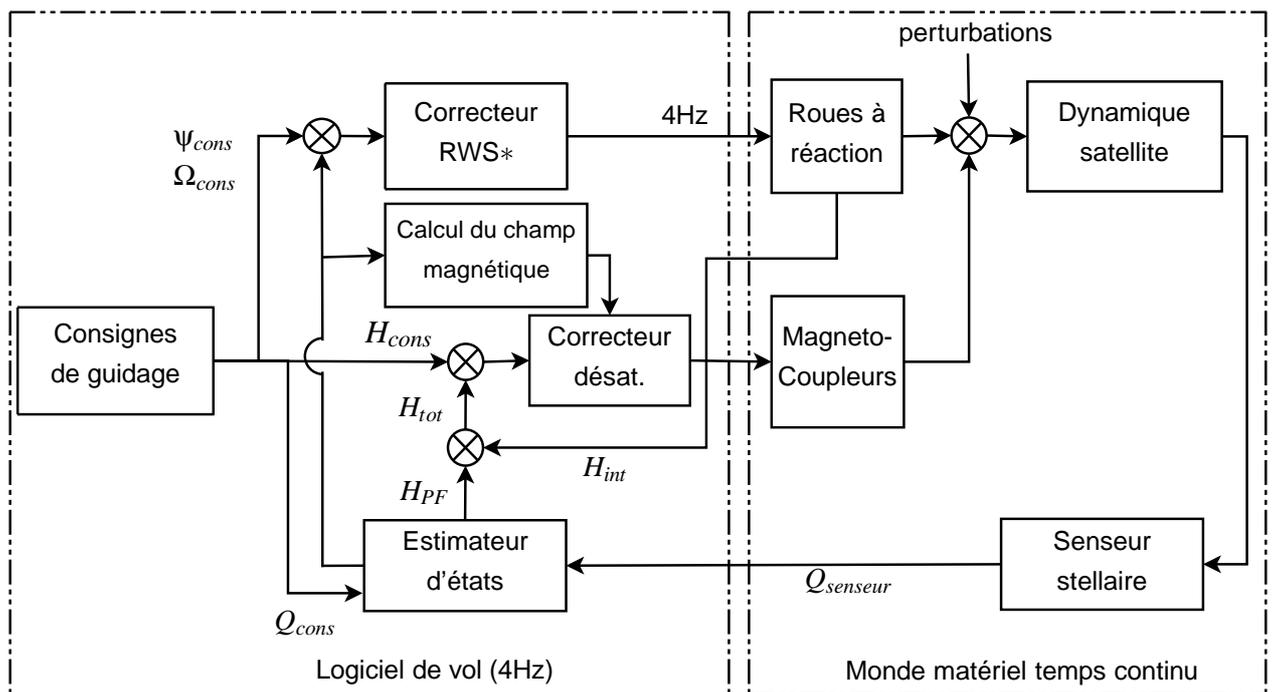


FIG. IV.2 – Boucle SCAO mode normal (\* roues à réaction)

- la cinématique du satellite  $\Omega(t)$  : c'est le vecteur rotation instantanée du repère satellite par rapport au repère inertiel. Il est exprimé dans le repère de consigne qui doit être maintenu aussi proche que possible du repère satellite par la boucle de pilotage.
- l'attitude du satellite  $Q(t)$  : elle est représentée par le quaternion associé à la rotation transformant le repère inertiel en un repère satellite.
- L'algorithme doit pouvoir réaliser des consignes pour les différents guidages que doit assurer la plate-forme microsatellite. Dans le cas que nous traitons, le guidage pourra être inertiel, géocentrique où encore héliocentrique. Selon le type de guidage, le calcul des profils de consigne se fait soit dans le repère inertiel soit dans le repère lié à l'orbite :
  - les profils de guidage dans le repère inertiel : la consigne de guidage téléchargée représente la cinématique du satellite par rapport au repère inertiel. Ce mode de guidage est utilisé pour les modes de pointage inertiel (satellite d'observation astronomique par exemple) et les rendez-vous d'attitude directement exprimés par le sol en repère inertiel.
  - les profils de guidage dans le repère orbital local ( $R_{ol}$ ) : la consigne de guidage téléchargée représente la cinématique du satellite par rapport au repère orbital local  $R_{ol}(t)$ . Ce mode de guidage est utilisé lorsque l'on veut asservir l'attitude du satellite à l'orbite. On trouve comme profil de guidage  $R_{ol}$  le pointage géocentrique.
- La sortie de l'algorithme doit être composée des paramètres qui seront téléchargés à bord à savoir les coefficients et les dates de validité du profil de vitesse ainsi que le quaternion initial. La vitesse courante est alors calculée en temps réel à bord et le quaternion courant est obtenu par intégration numérique de la vitesse courante.

Pour les manoeuvres que nous calculerons pour le microsatellite Demeter, la consigne sera calculée dans le  $R_{ol}$ . Les consignes de guidage d'attitude et d'orbite sont utilisées dans les

phases mode grossier de transition, mode de contrôle d'orbite et mode normal de vol.

### IV.3.2 Environnement et conditions de simulations

Toutes les simulations présentées dans les paragraphes suivants sont exécutées dans des conditions identiques. Nous simulons le satellite dans un mode normal de mission. Les manoeuvres sont définies dans le cadre d'un guidage géocentrique. Les générateurs solaires montés sur pivot motorisé sont fixes durant la manoeuvre, il n'y a donc pas de couples perturbateurs entraînés par leurs rotation. Les autres couples perturbateurs modélisés dans le simulateur, tels que ceux entraînés par la pression solaire, l'albédo terrestre, le gradient de gravité, les phénomènes aérodynamiques, moment résiduel magnétique créé par le satellite, sont pris en compte. On ne considèrera pas que les angles de garde qui définissent un secteur angulaire permettant le non éblouissement du senseur stellaire par la terre, le soleil ou la lune. De plus la mesure du senseur sera non bruité et toujours disponible. La bande passante de la boucle SCAO est fixé à  $0.015Hz$  sachant que la fréquence du premier mode souple est de  $0.4Hz$  dans le cas nominal. Nous inhiberons enfin la boucle de désaturation liée au magnéto-coupleurs.

### IV.3.3 Solutions calculées par platitude et collocation

Nous avons généré un ensemble de trajectoires en utilisant les méthodes du chapitre III. Notre premier objectif vise à comparer les temps optimum de transfert. Pour MANIAC, le temps optimal de transfert est d'environ 340 secondes. Pour notre générateur, il dépend de la configuration de la base des B-splines dans laquelle sont décrites les sorties plates. Le résultat est présenté dans la figure IV.3. Le temps minimal de transfert obtenu avec nos méthodes est de 295 secondes pour un polynôme par morceaux d'ordre 11 (ordre maximal supporté par le simulateur). De manière générale, nous obtenons pour la majeure partie des bases de B-splines des temps de transferts inférieurs à ceux produit par MANIAC.

### IV.3.4 Discussion et comparaison des résultats de simulation

Afin de procéder à une comparaison des profils générés par platitude et collocation et par MANIAC, nous nous plaçons dans des conditions d'ordre et de temps de manoeuvre identique. Ainsi les trajectoires des sorties plates sont décrites dans une base de B-splines d'ordre 6 sur 5 segments et les profils que nous simulerons sont donnés sur les figures IV.1 et IV.4. Nous proposons dans ce paragraphe la comparaison des simulations des profils générés par platitude et collocation (cf la figure IV.4) et par l'algorithme MANIAC (cf la figure IV.1). Les sorties du simulateur issues des profils générés par platitude et collocation et par MANIAC seront indicées par P.C. et par Ma. respectivement.

La figure IV.5 expose les trajectoires d'attitude en angles de Cardan. Il est à noter que la manoeuvre débute à  $50s$  et atteint la position finale à  $400s$ . La figure IV.6 relève l'écart entre la trajectoire nominale et la sortie d'attitude du simulateur. Les maxima en valeur absolue de ces écarts sont donnés dans le tableau IV.1 pour des profils d'ordre 6 à 11. Nous obtenons jusqu'à 40% de gain sur l'écart à la trajectoire pour des profils P.C. sur 350 secondes de temps de manoeuvre. Nous avons aussi observé que pour un ordre 11 sur un temps de manoeuvre de  $300s$ , nous avons des écarts équivalents à ceux obtenus par MANIAC pour  $350s$

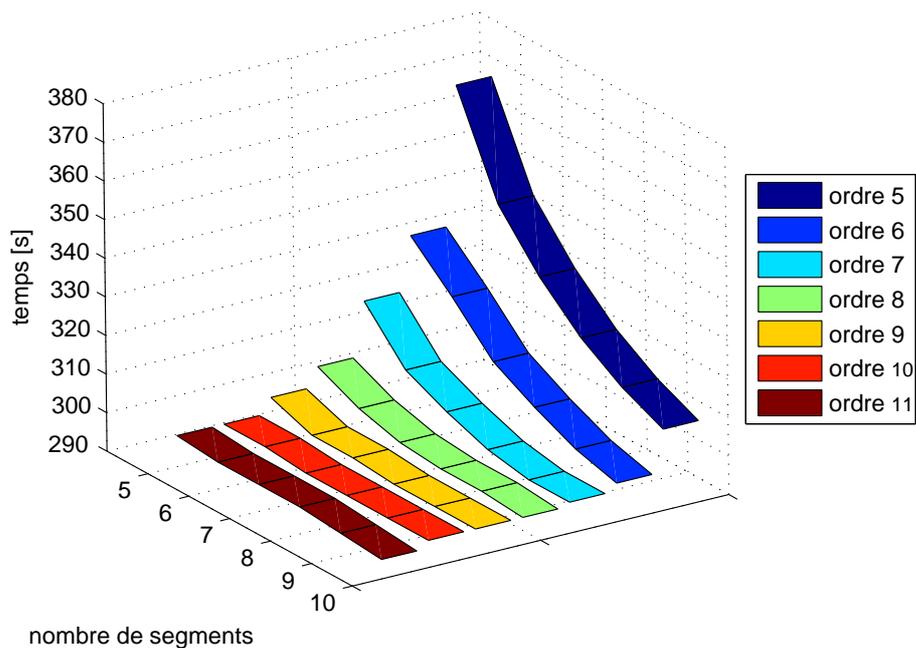


FIG. IV.3 – Temps de transfert optimal pour le satellite DEMETER en fonction de la base de B-splines

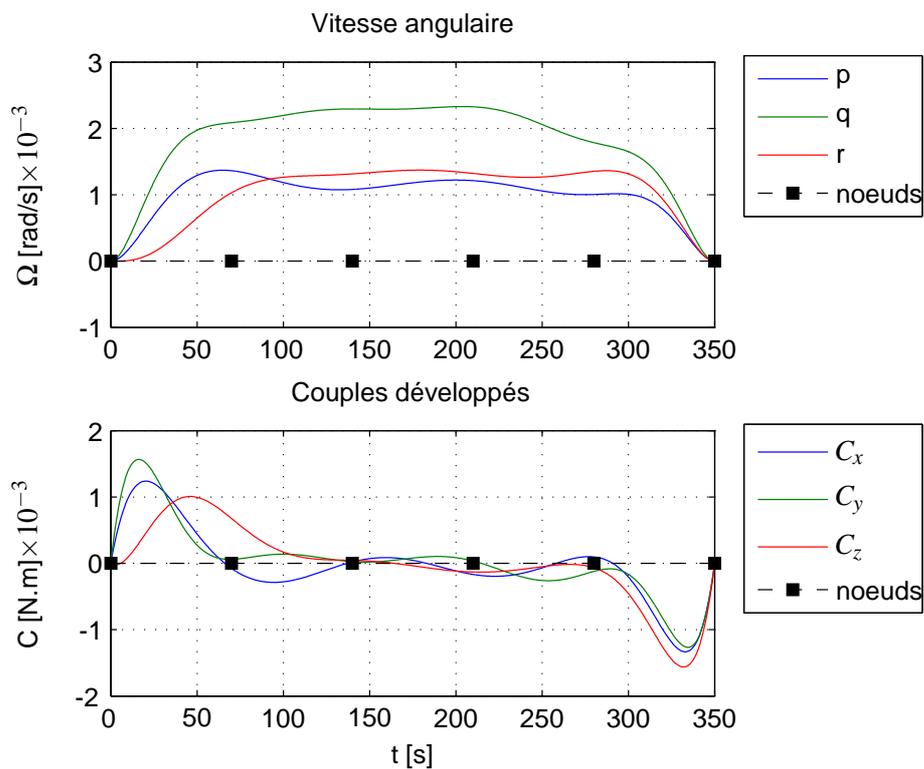


FIG. IV.4 – Profils de vitesse et de couple calculés par platitude pour une base de B-spline d'ordre 6

Les vitesses angulaires du satellite obtenues par le simulateur sont présentées en figure IV.7. La vitesse de rotation et la sortie de couples des roues à réaction sont exposés dans les

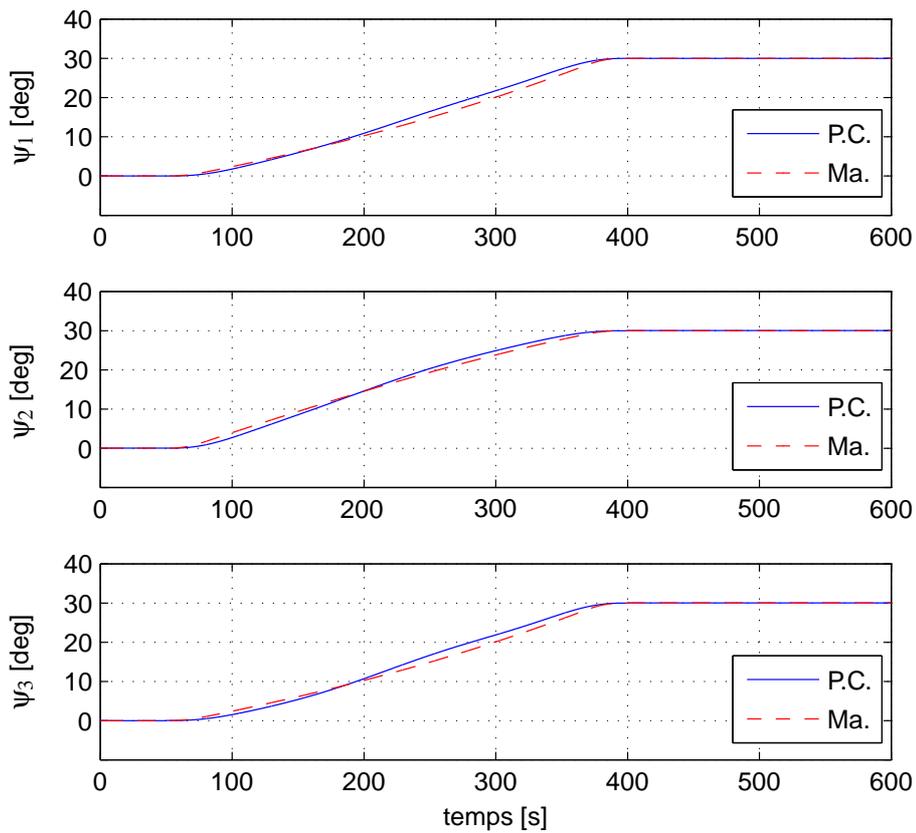


FIG. IV.5 – Trajectoire d'attitude simulée

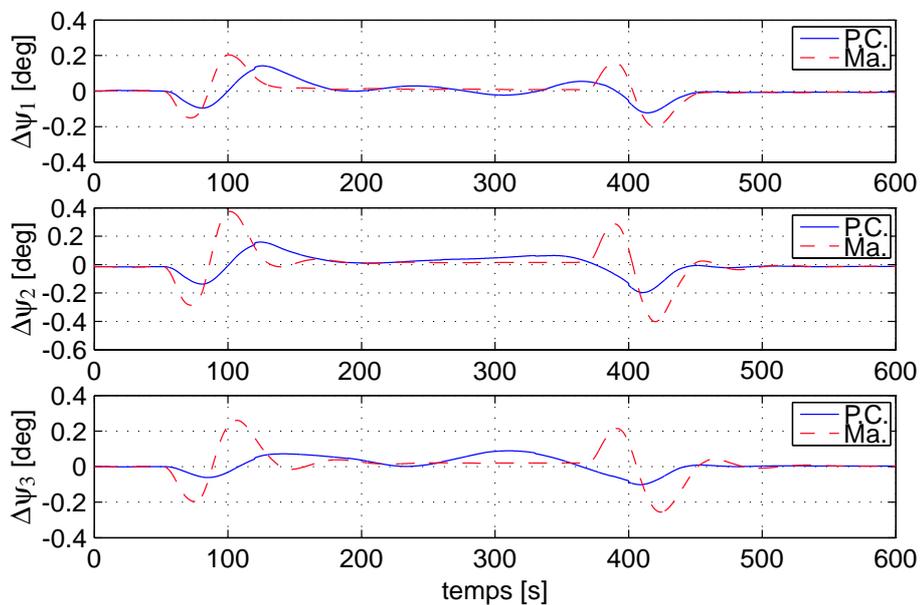


FIG. IV.6 – Ecart par rapport à la trajectoire d'attitude nominale

figures IV.8 et IV.9. On vérifie pour les deux profils que la vitesse des roues n'est jamais saturée durant la manoeuvre, la commandabilité du satellite est alors toujours assurée. Le couple est sensiblement moins utilisé lors de la simulation des profils P.C. Ceci s'explique par le fait que la

[deg]	Profils MANIAC			Profils P.C.				
	ordre 6 *	ordre 6 *	ordre 7 *	ordre 8 *	ordre 9 *	ordre 10 *	ordre 11*	ordre 11◇
$\psi_1$	0.20	0.14	0.15	0.16	0.16	0.18	0.19	0.24
$\psi_2$	0.40	0.24	0.24	0.26	0.27	0.29	0.29	0.34
$\psi_3$	0.26	0.16	0.17	0.18	0.18	0.20	0.20	0.20

TAB. IV.1 – Ecart maximum à la trajectoire nominale (\* : 350s, ◇ : 300s)

consommation de couple est le critère minimisé lors de la génération des profils. Le tableau IV.2 permet de comparer les efforts nécessaires pour relacher le véhicule après la manoeuvre. Bien que les couples utilisés pendant la phase de tranquillisation ne soient pas définis par les profils simulés, ils montrent un écart à la trajectoire nominale et une agitation des modes souples en fin de manoeuvre. Nous pouvons déduire que les profils P.C. entraînent sensiblement moins de perturbations que les profils MANIAC. Ce résultat est lié aux observations de la figure IV.6 qui montre bien que le temps de relaxation est plus court avec les profils P.C.

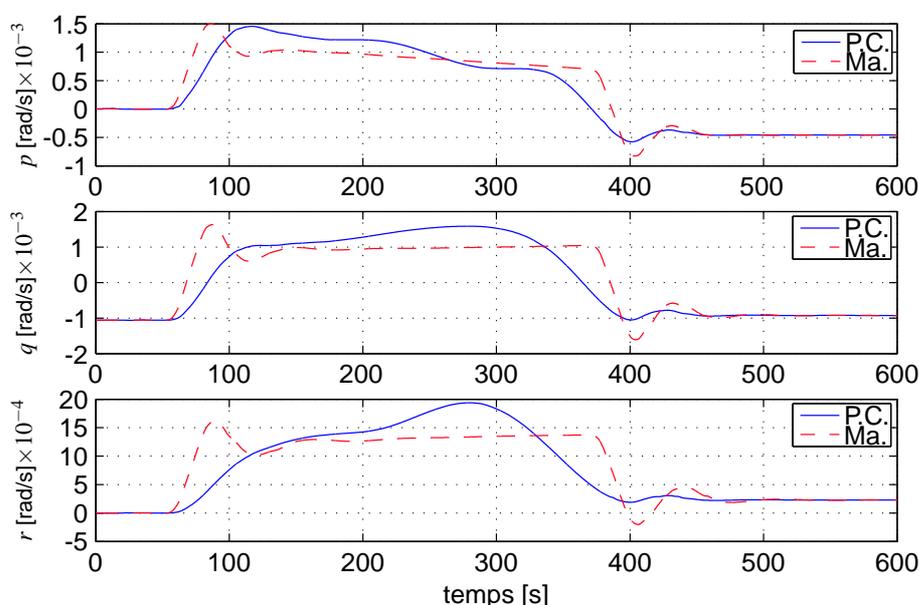


FIG. IV.7 – Vitesses angulaires du satellite

$10^{-3} [N.m.s]$	MANIAC	Profils P.C.							
	ordre	6 *	6 *	7 *	8 *	9 *	10 *	11*	11◇
$\int  torques $		116.0	58.6	61.0	82.1	96.6	100.2	101.4	112.0

TAB. IV.2 – Consommation du couple induit à la boucle de régulation lors de la phase de tranquillisation (\* : 350s, ◇ : 300s)

## IV.4 Conclusions

Nous avons réalisé dans ce chapitre une étude comparative des méthodes de génération de trajectoires développées dans le chapitre III et au CNES par le biais d'un simulateur dédié. Les

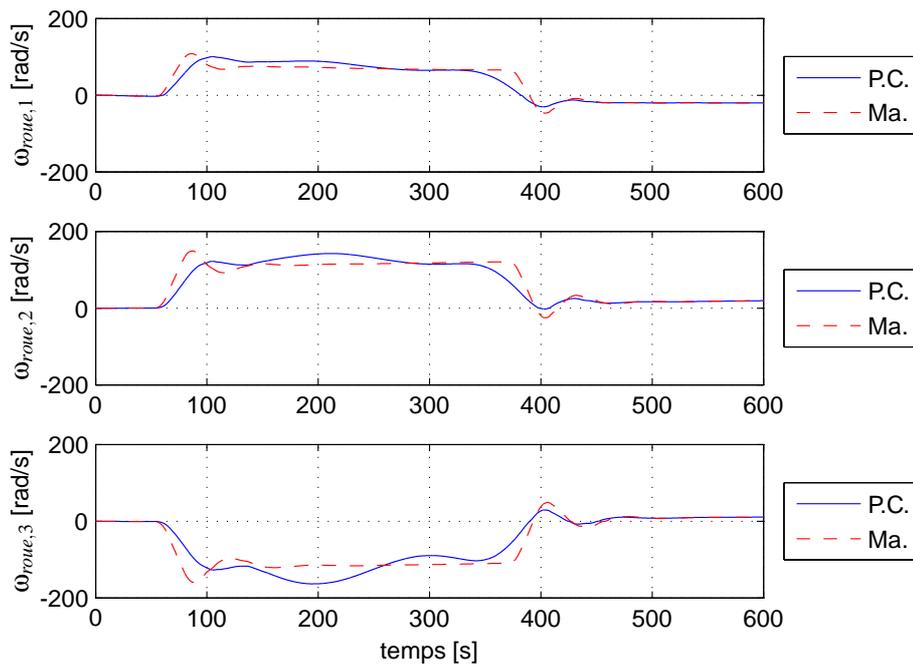


FIG. IV.8 – Vitesse de rotation des roues à réaction

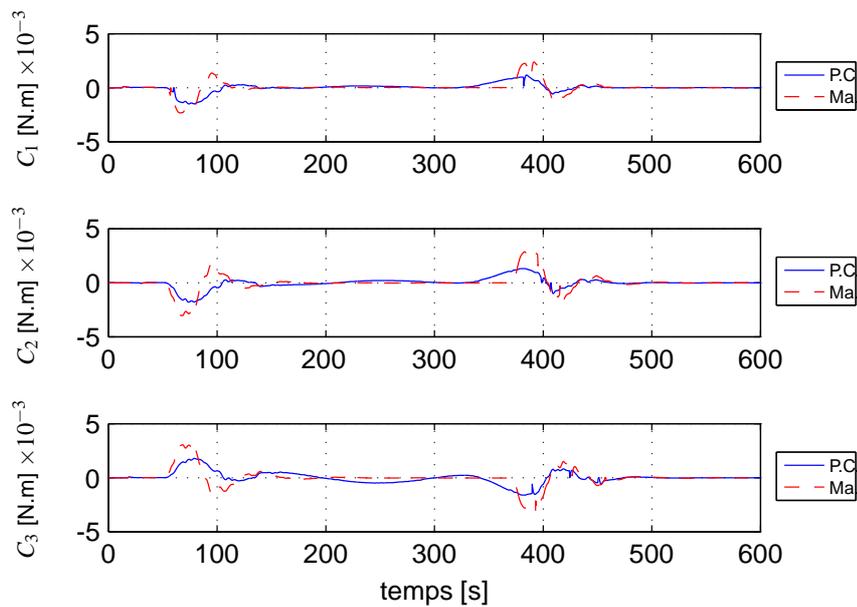


FIG. IV.9 – Couple de sorties des roues à réaction

résultats exposés montrent que l'utilisation de la platitude et de la collocation dans le problème de génération de trajectoires a de nombreux avantages par rapport aux méthodes mises en place dans MANIAC. En effet, Le temps de transfert optimal a été fortement réduit. De plus, sur une durée de manoeuvre équivalente, les profils P.C. permettent un gain significatif en termes de *tranquillisation* du satellite.



---

## Chapitre V

# Génération de trajectoires : formulation semi-définie positive

---

### Sommaire

---

<b>V.1</b>	<b>Introduction</b>	<b>96</b>
<b>V.2</b>	<b>Représentation en somme de carrés</b>	<b>96</b>
V.2.1	Cas des ordres impairs : $k_v = 2k_u - 1$	97
V.2.2	Cas des ordres pairs : $k_v = 2k_u$	101
<b>V.3</b>	<b>Application au problème de génération de trajectoires</b>	<b>102</b>
V.3.1	Appartenance d'une trajectoire $t \mapsto \bar{z}(t)$ à un demi-espace de $O_{\bar{z}}$	102
V.3.1.1	Formulation du problème de positivité	102
V.3.1.2	Base de représentation adjointe et opérateurs $\Lambda$ et $\Lambda^*$	104
V.3.1.3	Exemple	108
V.3.2	Appartenance d'une trajectoire $t \mapsto \bar{z}(t)$ à l'intersection de plusieurs demi-espaces de $O_{\bar{z}}$	110
V.3.2.1	Formulation du problème	110
V.3.2.2	Exemple	112
V.3.3	Application au problème de guidage	113
<b>V.4</b>	<b>Conclusions</b>	<b>114</b>

---

## V.1 Introduction

Afin de résoudre le problème de commande optimale, nous avons développé dans le chapitre III une première approche : à partir de méthodes numériques principalement basées sur la collocation par B-splines, nous avons transformé le problème initial en un problème de programmation non linéaire paramétrique. Nous avons pu voir que les contraintes sont satisfaites sur un ensemble de points temporels prédéfinis. En revanche, il n'y a aucune garantie qu'elles soient respectées sur des points n'appartenant pas à cet ensemble. Les simulations numériques ont clairement montré qu'une vérification *a posteriori* est souvent nécessaire pour s'assurer du respect des contraintes sur les instants qui se trouvent entre les points de collocation.

Dans ce chapitre, nous allons développer un cadre méthodologique novateur qui consiste à convertir notre problème, initialement formulé en termes de programmation semi-infinie, en un problème de programmation semi-définie positive. Cette approche aura comme avantage de vérifier les contraintes imposées sur le *continuum* temporel et non plus en un nombre d'instants fixés. La méthodologie développée est une extension des travaux réalisés par Henrion et Lasserre pour les systèmes linéaires plats [Henrion & Lasserre 2003]. Cependant, nos développements s'en distinguent par deux aspects. En premier lieu, nous nous intéressons à la classe des systèmes non linéaires plats. Ensuite, nous utilisons des B-splines, et non des polynômes, pour paramétrer les trajectoires des sorties plates.

La méthode que nous proposons repose sur la formulation d'un problème de génération de trajectoires sous contraintes linéaires en termes de positivité d'un polynôme par morceaux.

Ce chapitre sera structuré de la façon suivante :

Rappelons qu'un ensemble des définitions et des propriétés mathématiques relatives aux espaces de fonctions polynomiales par morceaux et aux B-splines est nécessaire à la compréhension de ce chapitre. Cet ensemble de pré-requis est disponible dans l'annexe C. Nous développerons ensuite une première contribution relative au théorème de représentation des polynômes par morceaux positifs sur le cône des matrices semi-définies positives. Puis, nous formulerons le problème de génération de trajectoires sous contraintes comme un problème d'appartenance des trajectoires des sorties plates à un polytope. Notons enfin que ce chapitre doit être considéré comme un chapitre d'ouverture, proposant les bases méthodologiques d'une nouvelle approche. Nous illustrerons par un exemple académique la mise en oeuvre de différents éléments techniques proposés. Nous présenterons également les résultats d'optimisation obtenus sur le modèle satellitaire utilisé au chapitre III, afin de mettre en évidence la faisabilité et l'applicabilité de l'approche.

## V.2 Représentation en somme de carrés

Dans cette section, nous allons établir un double théorème V.1 et V.2 qui nous permettra de qualifier l'ensemble des polynômes par morceaux positifs d'ordre arbitraire. Ce théorème définit le cône des coordonnées de ces derniers dans leur base de B-splines comme étant l'image du cône des matrices semi-définies positives. Pour ce faire, nous nous appuyerons sur les résultats obtenus par Nesterov relatifs à la représentation en somme de carrés des polynômes généralisés [Nesterov 2000].

Nous allons envisager ici deux cas distincts. Si  $k_v = 2k_u - 1$  i.e. le degré des polynômes par

morceaux est pair, il existe alors un ordre  $k_u$  arbitraire tel qu'un polynôme par morceaux appartenant à  $\mathbb{P}_{k_v, \xi}$  puisse s'exprimer comme une somme pondérée de carrés de polynômes par morceaux appartenant à  $\mathbb{P}_{k_u, \xi}$ . Si  $k_v = 2k_u$ , le raisonnement précédent n'est pas directement applicable. On utilisera alors une démarche légèrement différente. Etudions successivement ces deux cas.

### V.2.1 Cas des ordres impairs : $k_v = 2k_u - 1$

On s'intéresse à la somme de polynômes par morceaux élevés au carré et appartenant à  $\mathbb{P}_{k_u, \xi, v_u}$ . Le résultat d'une telle opération donne un polynôme par morceaux appartenant à  $\mathbb{P}_{2(k_u-1), \xi, v_u}$ . Posons  $\mathcal{S}_u$  la base des B-splines d'ordre  $k_u$  et de classes de continuité  $v_u = \{v_{u,1}, \dots, v_{u,l+1}\}$  associées à la séquence  $\xi$ . Alors  $\mathcal{S}_u$  est construite sur la séquence nodale  $T_u$  basée sur la séquence de segmentation  $\xi$  telle que

$$T_u = \left\{ \underbrace{\xi_1, \dots, \xi_1}_{\rho_{u,1} \text{ fois}}, \underbrace{\xi_2, \dots, \xi_2}_{\rho_{u,2} \text{ fois}}, \dots, \underbrace{\xi_{l+1}, \dots, \xi_{l+1}}_{\rho_{u,l+1} \text{ fois}} \right\} = \{t_1, \dots, t_{n_{T_u}}\}. \quad (\text{V.1})$$

La base de B-splines  $\mathcal{S}_u$  est une base du sous-espace de fonctions  $\mathbb{P}_{k_u, \xi, v_u}$  avec  $v_{u,i} = k_u - \rho_{u,i}$ ,  $i = 1, \dots, l+1$  telle que

$$\mathcal{S}_u = \{B_{1,k_u}, \dots, B_{m,k_u}\} = \{u_1(x), \dots, u_m(x)\} \quad (\text{V.2})$$

Notons  $\mathcal{F}(\mathcal{S}_u^2)$  le sous-espace défini par :

$$\mathcal{F}(\mathcal{S}_u^2) = \{P_\mu(x) = \sum_{i=1}^N \tau_i^2(x), \tau_i(x) \in \mathbb{P}_{k_u, \xi, v_u}, i = 1, \dots, N\} \quad (\text{V.3})$$

Par construction, le sous-espace de fonctions  $\mathcal{F}(\mathcal{S}_u^2)$  est défini sur la famille de fonctions  $\mathcal{S}_u^2$  formée par les produits deux à deux des éléments de la base  $\mathcal{S}$  i.e. :

$$\mathcal{S}_u^2 = \{u_i(x)u_j(x), i, j = 1, \dots, m\} \quad (\text{V.4})$$

La famille  $\mathcal{S}_u^2$  n'étant pas une base, nous allons maintenant définir une base  $v(x)$  dans laquelle tout élément de  $\mathcal{F}(\mathcal{S}_u^2)$  pourra être représenté.

**Proposition V.1** *Le sous-espace de fonctions  $\mathcal{F}(\mathcal{S}_u^2)$  admet comme base de représentation  $v(x)$ , une famille de  $n$  B-splines d'ordre  $2k_u - 1$  définie sur une séquence nodale  $T_v$  convenablement choisie telle que :*

$$\forall \mu(x) \in \mathcal{F}(\mathcal{S}_u^2), \quad P_\mu(x) = \sum_{i=1}^n \mu_i v_i(x) \quad (\text{V.5})$$

avec  $\mu = (\mu_1, \dots, \mu_n)$  et  $n$  le nombre d'éléments de la base  $v(x)$  tel que

$$n = \text{card}(T_v) - (2k_v - 1)$$

avec  $T_v$  définie par :

$$T_v = \left\{ \underbrace{\xi_1, \dots, \xi_1}_{\rho_{v,1} \text{ fois}}, \underbrace{\xi_2, \dots, \xi_2}_{\rho_{v,2} \text{ fois}}, \dots, \underbrace{\xi_{l+1}, \dots, \xi_{l+1}}_{\rho_{v,l+1} \text{ fois}} \right\} = \{t_1, \dots, t_{n_{T_v}}\}. \quad (\text{V.6})$$

où  $\rho_{v,i} = 2k_v - 1 - v_{u,i}$ . ■

**Démonstration**

Commençons par décrire les caractéristiques des éléments de  $\mathcal{S}_u^2$ .  $\mathcal{S}_u$  étant la base des B-splines d'ordre  $k_u$ , les éléments de cette base sont des fonctions polynomiales de degré  $k_u - 1$  par morceaux. Il vient que les éléments de  $\mathcal{S}_u^2$  sont des polynômes par morceaux de degré  $2(k_u - 1)$ . La classe de continuité  $v_{u,i}$  des fonctions de  $\mathcal{F}(\mathcal{S}_u)$  est infinie sauf en  $\{\xi_i\}_{i=1,\dots,l+1}$  où elle est  $k_u - p_i$ . La classe de continuité des éléments  $\mathcal{F}(\mathcal{S}_u^2)$  est donc caractérisée par :

$$\forall P_\mu(x) \in \mathcal{F}(\mathcal{S}_u^2), P_\mu(x) \text{ est de classe } C^{v_{u,i}} \text{ si } x \in \xi \quad (\text{V.7})$$

Ainsi par construction,  $\mathcal{F}(\mathcal{S}_u^2) \subseteq \mathbb{P}_{2k_u-1, \xi, v_u}$ . Or, d'après le théorème de Curry et Schoenberg [Curry & Schoenberg 1966],  $\mathbb{P}_{2k_u-1, \xi, v_u}$  admet comme base une famille de B-splines  $\mathcal{S}_v$  d'ordre  $k_v = 2k_u - 1$  construite sur la séquence  $T_v$  définie précédemment :

$$\mathcal{S}_v = \{B_{1,k_v}, \dots, B_{n,k_v}\} = \{v_1(x), \dots, v_n(x)\} \quad (\text{V.8})$$

□

Nous allons maintenant caractériser  $\mathcal{F}(\mathcal{S}_u^2)$  à l'aide de la notion de cône dual associée au cône des moments <sup>1</sup> de la base  $v(x)$  des éléments de ce sous-espace.

**Définition V.1 (cône des moments)**

Le cône des moments de la base des B-splines  $\mathcal{S}_v$  se définit par :

$$\mathbb{M} = \{c = (c_1, \dots, c_n) : c_i = \int_{\Delta} v_i(x) d\sigma(x), i = 1, \dots, n\} \quad (\text{V.11})$$

où la mesure  $\sigma(x)$  est une fonction continue, bornée et non décroissante sur  $\Delta$ .

**Définition V.2 (cône dual)**

On note par  $\mathbb{K}$  le cône dual du cône  $\mathbb{M}$  décrit par :

$$\mathbb{K} = \{\mu = (\mu_1, \dots, \mu_n) : \sum_{i=1}^n \mu_i c_i \geq 0, \quad \forall c \in \mathbb{M}\} \quad (\text{V.12})$$

Il a été prouvé dans le cadre des systèmes de Tchebychev [Karlin & Studden 1966, théorème 9.1] puis dans un contexte élargi englobant les B-splines [Nesterov 2000] que tout élément  $\mu$  du cône  $\mathbb{K}$  détermine un polynôme par morceaux  $P_\mu(x) = \sum_{i=1}^n \mu_i v_i(x)$  tel que :

$$P_\mu(x) \geq 0, \quad \forall x \in \Delta \quad (\text{V.13})$$

Nous allons par la suite décrire les coordonnées des polynômes par morceaux définis sur la famille de fonctions  $\mathcal{S}_u^2$  dans la base  $v(x)$ .

<sup>1</sup> Soient une fonction  $f : I \rightarrow \mathbb{R}$  continue sur un intervalle  $I$  (non réduit à un point) de  $\mathbb{R}$  et une mesure  $\sigma(x)$ , une fonction continue, bornée et non décroissante sur  $I$ . On définit le moment de  $f$  (sous réserve d'existence) par :

$$m_\sigma(f) = \int_I f(x) d\sigma(x) \quad (\text{V.9})$$

En particulier, étant donné un entier naturel  $n$ , on définit le moment d'ordre  $n$  de  $f$  (sous réserve d'existence) par :

$$m_n(f) = \int_I x^n f(x) dx \quad (\text{V.10})$$

Posons la notation  $\langle \cdot, \cdot \rangle$  qui définit le produit interne tel que pour  $A$  et  $B$ , deux matrices de  $\mathbb{R}^{\alpha \times \beta}$  nous ayons :

$$\mathbb{R}^{\alpha \times \beta} \times \mathbb{R}^{\alpha \times \beta} \rightarrow \mathbb{R}, A, B \mapsto \langle A, B \rangle = \sum_{i=1}^{\alpha} \sum_{j=1}^{\beta} a_{i,j} b_{i,j}$$

Ce produit revient à faire la somme des produits terme à terme des deux matrices ou vecteurs. Introduisons maintenant l'opérateur linéaire matriciel  $\Lambda(v)$  qui associe à la base  $v(x)$  de  $\mathbb{P}_{k_v, \xi, v_v}$  une matrice  $m \times m$  de polynômes par morceaux de  $\mathbb{P}_{k_v, \xi, v_v}$

$$\mathbb{P}_{k_v, \xi, v_v}^n \rightarrow \mathbb{P}_{k_v, \xi, v_v}^{m \times m} : v(x) \mapsto \Lambda(v) \equiv u(x)u(x)^T, \quad x \in \Delta \quad (\text{V.15})$$

en notant  $u(x) = (u_1(x), \dots, u_m(x))^T$ . On définit  $\lambda_{i,j} \in \mathbb{R}^n$  les coordonnées dans la base  $v(x)$  du terme  $(i, j)$  de la matrice  $\Lambda(v)$  telle que :

$$\Lambda(v) = \begin{bmatrix} \langle \lambda_{1,1}, v \rangle & \dots & \langle \lambda_{1,m}, v \rangle \\ \vdots & \ddots & \vdots \\ \langle \lambda_{m,1}, v \rangle & \dots & \langle \lambda_{m,m}, v \rangle \end{bmatrix} = (u_1(x), \dots, u_m(x))(u_1(x), \dots, u_m(x))^T \quad (\text{V.16})$$

Une combinaison linéaire des termes de la matrice  $\Lambda(v)$  s'écrit :

$$\langle Y, \Lambda(v) \rangle, \quad Y \in \mathbb{R}^{m \times m}$$

Définissons l'opérateur linéaire dual  $\mathbb{R}^{m \times m} \rightarrow \mathbb{R}^n, Y \mapsto \Lambda^*(Y)$  tel que :

$$\langle Y, \Lambda(v(x)) \rangle \equiv \langle \Lambda^*(Y), v(x) \rangle, \quad \forall Y \in \mathbb{R}^{m \times m}, v \in \mathbb{R}^n \quad (\text{V.17})$$

Cette dernière relation établit un lien entre les pondérations sur  $\mathcal{S}_u^2$ , définissant un polynôme par morceaux, et les coordonnées dans la base  $v(x)$  de ce même polynôme. La détermination des opérateurs  $\Lambda(v)$  et  $\Lambda^*(Y)$  dépend uniquement des bases  $u(x)$  et  $v(x)$ . La méthode de calcul sera détaillée par la suite. Nous pouvons maintenant établir le théorème qui définit le sous espace des fonctions polynomiales par morceaux positives ou nulles par le biais d'une condition de positivité semi-définie sur la matrice  $Y$ . Nous pouvons maintenant poser établir la condition de positivité d'un polynôme par morceaux.

**Théorème V.1** Soit  $\mu$  un élément du cône  $K$  convexe pointé et fermé défini par :

$$K = \{\mu \in \mathbb{R}^n : \mu = \Lambda^*(Y), Y \succeq 0\} \quad (\text{V.18})$$

A chaque élément du cône  $K$  est associé une fonction  $P_\mu(x) = \langle \mu, v(x) \rangle$ , est définie comme une somme de carrés telle que :

$$P_\mu(x) = \sum_{i=1}^n \mu_i v_i(x) = \sum_{i=1}^N \tau_i^2(x) \geq 0, \quad \tau_i(x) \in \mathbb{P}_{k_u, \xi, v_u}, \quad i = 1, \dots, N \quad (\text{V.19})$$

■

**Démonstration**

Soient  $P_\mu(x) = \langle \mu, v(x) \rangle$  où  $\mu = \Lambda^*(Y)$  avec  $Y \succeq 0$ . Ainsi, on peut écrire  $\forall x \in \Delta$  :

$$P_\mu(x) = \langle \Lambda^*(Y), v(x) \rangle = \langle Y, \Lambda(v(x)) \rangle \quad (\text{V.20})$$

$$= \langle Y, u(x)u(x)^T \rangle \quad (\text{V.21})$$

avec  $\langle Iu(x), u(x) \rangle = \langle I, u(x)u(x)^T \rangle$  et  $I$  la matrice identité, il vient :

$$= \langle Yu(x), u(x) \rangle = (Yu(x))^T u(x) \quad (\text{V.22})$$

$$= u(x)^T Y^T u(x) \geq 0 \quad (\text{V.23})$$

D'autre part, si  $P_\mu(x) \geq 0$ , alors il existe un jeu de coefficients  $\gamma_i \in \mathbb{R}^m, i = 1, \dots, N$  tel que :

$$P_\mu(x) = \sum_{i=1}^N \langle \gamma_i, u(x) \rangle^2 = \left\langle \sum_{i=1}^N \gamma_i \gamma_i^T, u(x)u(x)^T \right\rangle \quad (\text{V.24})$$

$$= \left\langle \sum_{i=1}^N \gamma_i \gamma_i^T, \Lambda(v(x)) \right\rangle = \left\langle \Lambda^* \left( \sum_{i=1}^N \gamma_i \gamma_i^T \right), v(x) \right\rangle \quad (\text{V.25})$$

□

Ainsi on en déduit que la matrice  $Y$  peut être choisie de manière à avoir

$$Y = \sum_{i=1}^N \gamma_i \gamma_i^T \succeq 0$$

**Remarque V.1** [Nesterov 2000] Il est toujours possible de décrire le cône  $M$  en termes d'inégalités matricielles linéaires lorsque les fonctions  $P_\mu(x)$  sont représentables comme des sommes de carrés :

$$M = \{c \in \mathbb{R}^n : A(c) \succeq 0, i = 1, \dots, k\} \quad (\text{V.26})$$

avec les matrices  $A$  dépendant linéairement de  $c$ . Une telle description du cône  $M$  peut être déduite de l'expression (V.17) et de la définition du cône dual de  $K$  (V.18) :

$$c \in M \Leftrightarrow \exists \mu \in K, \langle \mu, c \rangle \geq 0 \quad (\text{V.27})$$

$$\Leftrightarrow \exists Y \succeq 0, \langle \Lambda^*(Y), c \rangle \geq 0 \quad (\text{V.28})$$

$$\Leftrightarrow \exists Y \succeq 0, \langle Y, \Lambda(c) \rangle \geq 0 \quad (\text{V.29})$$

$$\Leftrightarrow \Lambda(c) \succeq 0 \quad (\text{V.30})$$

Notons que  $M$  n'est pas vide car  $K$  est un cône pointé.

Nous venons de traiter le cas où la base  $v(x)$  est d'ordre impair. Le cas de l'ordre pair nécessite un traitement spécifique qui renvoie à la notion de somme de carrés pondérés définie par Nesterov [Nesterov 2000].

### V.2.2 Cas des ordres pairs : $k_v = 2k_u$

Reprenons  $\mathcal{S}_u$  la base des B-splines d'ordre  $k_u$  telle qu'elle a été décrite dans la section précédente. Nous nous intéressons cette fois à la description du sous-espace de fonctions  $\mathcal{F}(\bar{\mathcal{S}}_u^2)$  :

$$\mathcal{F}(\bar{\mathcal{S}}_u^2) = \{P_\mu(x) = \bar{\tau}(x) \sum_{i=1}^N \tau_i^2(x), \tau_i(x) \in \mathbb{P}_{k_u, \xi, v_u}, i = 1, \dots, N\} \quad (\text{V.31})$$

où  $\bar{\tau}(x)$  est un polynôme de degré 1 tel que

$$\bar{\tau}(x) > 0 \quad \forall x \in \Delta \quad (\text{V.32})$$

Définissons la famille  $\bar{\mathcal{S}}_u$  à partir de la base de B-splines  $\mathcal{S}_u$  :

$$\bar{\mathcal{S}}_u = \{\bar{u}_1 = \sqrt{\bar{\tau}(x)}u_1(x), \dots, \bar{u}_m = \sqrt{\bar{\tau}(x)}u_m(x)\} \quad (\text{V.33})$$

$\mathcal{S}_u$  étant une base libre, on déduit que  $\bar{\mathcal{S}}_u$  est aussi une base libre. Le sous-espace de fonctions  $\mathcal{F}(\bar{\mathcal{S}}_u^2)$  est construit sur  $\bar{\mathcal{S}}_u^2$ , l'ensemble des fonctions carrées de la base  $\bar{\mathcal{S}}_u$  :

$$\bar{\mathcal{S}}_u^2 = \{\bar{u}_i(x)\bar{u}_j(x), \quad i, j = 1, \dots, m\} \quad (\text{V.34})$$

Grâce à la proposition V.1, nous pouvons affirmer qu'il existe une base de B-splines  $v(x) = (v_1(x), \dots, v_n(x))^T$  d'ordre  $k_v$  pair permettant de recouvrir le sous-espace  $\mathcal{F}(\bar{\mathcal{S}}_u^2)$  engendrée par  $\bar{\mathcal{S}}_u^2$  car  $\mathcal{F}(\bar{\mathcal{S}}_u^2) \subseteq \mathbb{P}_{2k_u, \xi, v_u}$ .

À l'instar du cas impair, nous allons pouvoir décrire les éléments de  $\mathbb{P}_{2k_u, \xi, v_u}$  appartenant au sous-espace de fonctions  $\mathcal{F}(\bar{\mathcal{S}}_u^2)$  par l'écriture du cône  $\mathbb{K}(\bar{\tau})$ .

Définissons l'opérateur matriciel linéaire  $\Lambda(v)$  tel que :

$$\mathbb{P}_{k_v, \xi, v_v}^n \rightarrow \mathbb{P}_{k_v, \xi, v_v}^{m \times m} : v(x) \mapsto \Lambda(v(x)) \equiv \bar{u}(x)\bar{u}(x)^T, \quad x \in \Delta \quad (\text{V.35})$$

en notant  $\bar{u}(x) = (\bar{u}_1(x), \dots, \bar{u}_m(x))^T$  il vient :  $\forall x \in \Delta$ ,

$$\Lambda(v) = \begin{bmatrix} \langle \lambda_{1,1}, v \rangle & \dots & \langle \lambda_{1,m}, v \rangle \\ \vdots & \ddots & \vdots \\ \langle \lambda_{m,1}, v \rangle & \dots & \langle \lambda_{m,m}, v \rangle \end{bmatrix} = (\bar{u}_1(x), \dots, \bar{u}_m(x))^T (\bar{u}_1(x), \dots, \bar{u}_m(x)) \quad (\text{V.36})$$

Définissons maintenant l'opérateur linéaire dual  $\mathbb{R}^{m \times m} \rightarrow \mathbb{R}^n, Y \mapsto \Lambda^*(Y)$  tel que :

$$\langle Y, \Lambda(v(x)) \rangle \equiv \langle \Lambda^*(Y), v(x) \rangle, \quad \forall Y \in \mathbb{R}^{m \times m}, v \in \mathbb{R}^n \quad (\text{V.37})$$

Posons maintenant la condition de positivité polynôme par morceaux d'ordre pair.

**Théorème V.2** Soit  $\mu$  un élément du cône  $\mathbb{K}$  convexe pointé et fermé défini par :

$$\mathbb{K} = \{\mu \in \mathbb{R}^n : \mu = \Lambda^*(Y), Y \succeq 0\} \quad (\text{V.38})$$

À chaque élément du cône  $\mathbb{K}$  est associé une fonction  $P_\mu(x) = \langle \mu, v(x) \rangle$  est définie comme une somme de carrés pondérée par  $\bar{\tau}(x)$ , un polynôme de degré 1 strictement positif sur  $\Delta$ , tel que :

$$P_\mu(x) = \sum_{i=1}^n \mu_i v_i(x) = \bar{\tau}(x) \sum_{i=1}^N \tau_i^2(x) \geq 0, \quad \tau_i(x) \in \mathbb{P}_{k_u, \xi, v_u}, \quad i = 1, \dots, N \quad (\text{V.39})$$

■

**Démonstration**

Posons  $\mu(x) = \langle \mu, v(x) \rangle$  où  $\mu = \Lambda^*(Y)$  et  $Y \succeq 0$ . Ainsi, on peut écrire  $\forall x \in \Delta$  :

$$\begin{aligned} \mu(x) &= \langle \Lambda^*(Y), v(x) \rangle = \langle Y, \Lambda(v(x)) \rangle \\ &= \langle Y, \bar{u}(x)\bar{u}(x)^T \rangle = \langle Y\bar{u}(x), \bar{u}(x) \rangle \geq 0 \end{aligned} \quad (\text{V.40})$$

D'autre part, si  $\mu(x) \in K$ , alors il existe un jeu de coefficients  $\gamma_i \in \mathbb{R}^m, i = 1, \dots, N$  tel que :

$$\begin{aligned} \mu(x) &= \sum_{i=1}^N \langle \gamma_i, \bar{u}(x) \rangle^2 = \langle \sum_{i=1}^N \gamma_i \gamma_i^T, \bar{u}(x)\bar{u}(x)^T \rangle \\ &= \langle \sum_{i=1}^N \gamma_i \gamma_i^T, \Lambda(v(x)) \rangle = \langle \Lambda^* \left( \sum_{i=1}^N \gamma_i \gamma_i^T \right), v(x) \rangle \end{aligned} \quad (\text{V.41})$$

□

Ainsi on en déduit que la matrice  $Y$  peut être choisie telle que

$$Y = \sum_{i=1}^N \gamma_i \gamma_i^T \succeq 0$$

Nous disposons maintenant de conditions permettant de conclure à la positivité d'un polynôme par morceaux d'ordre arbitraire. Nous allons replacer cette première contribution dans le contexte de la génération de trajectoires.

**V.3 Application au problème de génération de trajectoires**

Deux contributions seront ici développées. Nous aborderons tout d'abord le problème du positionnement d'une trajectoire paramétrée par des B-splines,  $t \mapsto \bar{z}(t)$  par rapport à un hyperplan de l'espace  $O_{\bar{z}}$ , en termes de positivité d'un polynôme par morceaux. Puis, nous écrivons le problème d'appartenance d'une trajectoire de sorties plates  $t \mapsto \bar{z}(t)$  à un polytope de l'espace  $O_{\bar{z}}$  comme une somme de problèmes de positivité. A cette occasion, nous appliquerons notre contribution au problème de guidage.

**V.3.1 Appartenance d'une trajectoire  $t \mapsto \bar{z}(t)$  à un demi-espace de  $O_{\bar{z}}$** 

Dans les paragraphes suivants, nous allons traduire le problème d'une contrainte linéaire sur  $\bar{z}$  en un problème de positivité d'une B-spline dont les coefficients sont linéairement dépendants des paramètres des trajectoires de sortie plates. Ces coefficients seront alors déterminés à l'aide des théorèmes V.1 et V.2, ceci garantira l'appartenance de  $\bar{z}$  au demi-espace souhaité.

**V.3.1.1 Formulation du problème de positivité**

Pour l'utilisation des théorèmes de représentation des polynômes par morceaux positifs sur le cône des matrices semi-définies positives, nous nous placerons dans le cadre d'un exemple lié à la génération de trajectoires pour système plat.

Soit un espace de sorties plates  $O_{\bar{z}}$  de dimension  $n_{\bar{z}}$  dont les coordonnées sont les composantes du vecteur  $\bar{z}(t)$

$$\bar{z} = (z_1, \dots, z_{m_z}, \dot{z}_1, \dots, \dot{z}_{m_z}, z_1^{(r)}, \dots, z_{m_z}^{(r)})^T$$

La trajectoire  $\Delta = [t_0, t_f] \in \mathbb{R} \rightarrow \mathbb{R}^{n_{\bar{z}}}$ ,  $t \mapsto \bar{z}(t)$  doit appartenir à un demi-espace ouvert de  $O_{\bar{z}}$  délimité par un hyperplan. L'hyperplan est défini par  $a \in \mathbb{R}^{n_{\bar{z}}}$  et  $b \in \mathbb{R}$ . La contrainte d'appartenance à un demi-espace s'écrit alors :

$$\langle a, \bar{z} \rangle \geq b \quad (\text{V.42})$$

Nous considérons que les trajectoires des sorties plates sont représentées dans la base des B-splines d'ordre  $k$  :

$$z_i(t) = \sum_{j=1}^n C_{i,j} B_{j,k}(t), \quad i = 1, \dots, m_z \quad (\text{V.43})$$

Ainsi la  $r^{\text{ème}}$  dérivée de ces sorties plates s'écrit :

$$z_i^{(r)}(t) = \sum_{j=1}^n C_{i,j} B_{j,k}^{(r)}(t), \quad i = 1, \dots, m_z \quad (\text{V.44})$$

Nous allons maintenant développer la contrainte (V.42) :

$$\begin{aligned} \langle a, \bar{z} \rangle &\geq b \\ \Leftrightarrow a_1 z_1 + \dots + a_{m_z} z_{m_z} + a_{m_z+1} \dot{z}_1 + \dots + a_{2m_z} \dot{z}_{m_z} + \dots \\ &\quad + a_{(r-1)m_z+1} z_1^{(r)} + \dots + a_{m_z r} z_{m_z}^{(r)} \geq b \\ \Leftrightarrow \sum_{j=1}^n (a_1 C_{1,j} + \dots + a_{m_z} C_{m_z,j}) B_{j,k}(t) \dots \\ &\quad + (a_{m_z+1} C_{1,j} + \dots + a_{2m_z} C_{m_z,j}) \dot{B}_{j,k}(t) + \dots \\ &\quad + (a_{m_z(r-1)+1} C_{1,j} + \dots + a_{m_z r} C_{m_z,j}) B_{j,k}^{(r)}(t) \geq b \end{aligned} \quad (\text{V.45})$$

$$\begin{aligned} \Leftrightarrow \sum_{j=1}^n (a_1 B_{j,k}(t) + \dots + a_{m_z(r-1)+1} B_{j,k}^{(r)}(t)) C_{1,j} \dots \\ &\quad + (a_2 B_{j,k}(t) + \dots + a_{m_z(r-1)+2} B_{j,k}^{(r)}(t)) C_{2,j} + \dots \\ &\quad + (a_{m_z} B_{j,k}(t) + \dots + a_{m_z r} B_{j,k}^{(r)}(t)) C_{m_z,j} \geq b \end{aligned} \quad (\text{V.46})$$

Remarquons que l'inégalité (V.45) est une contrainte de positivité de la somme pondérée d'un polynôme par morceaux et de ses  $r$  dérivées successives. La contrainte (V.46) qui est équivalente à la contrainte (V.42) va permettre de réécrire le problème avec une seule base de B-splines.

**Proposition V.2** La somme d'un polynôme par morceaux appartenant à  $\mathbb{P}_{k,\xi,v}$  et de ses  $r$  dérivées successives est représentable dans une base de B-splines  $\beta_{i,k} \in \mathbb{P}_{k,\xi,v \ominus r}$ . ■

### Démonstration

On sait que  $B_{j,k} \in \mathbb{P}_{k,\xi,v}$ . On en déduit que  $\dot{B}_{j,k} \in \mathbb{P}_{k-1,\xi,v \ominus 1}$ ,  $\ddot{B}_{j,k} \in \mathbb{P}_{k-2,\xi,v \ominus 2}$ ,  $B_{j,k}^{(r)} \in \mathbb{P}_{k-2,\xi,v \ominus r}$ . Il vient que de la somme pondérée d'une B-spline  $B_{j,k}$  et de ses  $r$  dérivées successives appartient à  $\mathbb{P}_{k,\xi,v \ominus r}$ . L'opérateur  $\ominus$  est défini par :

$$v \ominus r = (v_1, v_2 - r, \dots, v_l - r, v_{l+1}), \quad r \leq v_i, i = 2, \dots, l \quad (\text{V.47})$$

Or d'après le théorème de Curry-Schoenberg [Curry & Schoenberg 1966], il existe une base  $\beta_{i,k}$  construite sur une séquence  $T_\beta$  convenablement définie. □

Ainsi l'inégalité (V.46) peut se reformuler par :

$$\sum_{j=1}^n \left( \left( \sum_{i=1}^{n_\beta} \alpha_{1,i,j} \beta_{i,k}(t) \right) C_{1,j} + \dots + \left( \sum_{i=1}^{n_\beta} \alpha_{m_z,i,j} \beta_{i,k}(t) \right) C_{m_z,j} \right) \geq b \quad (\text{V.48})$$

où  $\alpha_{p,j} \in \mathbb{R}^{n_\beta}$   $p = 1, \dots, m_z$  et  $j = 1, \dots, n$ . En mettant en facteur  $\beta_{i,k}$ , il vient

$$\sum_{i=1}^{n_\beta} \left( \sum_{j=1}^n \alpha_{1,i,j} C_{1,j} + \dots + \alpha_{m_z,i,j} C_{m_z,j} \right) \beta_{i,k}(t) \geq b \quad (\text{V.49})$$

On obtient alors :

$$\kappa(t) = \sum_{i=1}^{n_\beta} \kappa_i \beta_{i,k}(t) \geq 0 \quad (\text{V.50})$$

où

$$\kappa_i = \sum_{j=1}^n (\alpha_{1,i,j} C_{1,j} + \dots + \alpha_{m_z,i,j} C_{m_z,j}) - b \quad (\text{V.51})$$

**Remarque V.2** *Le paramètre  $b$  peut être une fonction polynomiale par morceaux du temps. Cette fonction polynomiale par morceaux doit cependant être représentable dans la base  $\beta$ .*

Nous venons ainsi de formuler le problème d'appartenance d'une trajectoire  $t \mapsto \bar{z}(t)$  à un demi-espace de l'espace  $O_{\bar{z}}$  en un problème de positivité d'une B-spline (V.50). Afin d'appliquer le théorème de positivité à ce problème, il nous faut déterminer les base  $u$  et  $v$  et les opérateurs  $\Lambda$  et  $\Lambda^*$ .

### V.3.1.2 Base de représentation adjointe et opérateurs $\Lambda$ et $\Lambda^*$

Il nous faut définir une base de B-splines  $u$  telle que

$$\mathcal{F}(S_u^2) \in \mathbb{P}_{k,\xi,v \ominus r}$$

On en déduit que la base  $u$  est d'ordre  $k_u$  telle que :

$$k_u = \begin{cases} \frac{k+1}{2} & \text{si } k \text{ est impair} \\ \frac{k}{2} & \text{si } k \text{ est pair} \end{cases} \quad (\text{V.52})$$

De plus, si l'on fait l'hypothèse que la base  $B_{j,k}$  est de continuité maximale à savoir  $v_i = k - 1$ ,  $i = 2, \dots, l$  et discontinue en  $\xi_1$  et  $\xi_{l+1}$ , il vient alors que les éléments de  $\mathcal{F}(S_u^2)$  sont de classe  $C^{k-r-1}$  en  $\xi_i$ ,  $i = 2, \dots, l$ .

Si  $r < k_u$ , cela implique alors que la classe de continuité de la base  $u$  soit supérieure à son ordre, ce qui est contraire à la définition des B-splines qui impose  $k > v_i$  pour tout  $i$  (cf. equation (C.7)). Le cas  $r \geq k_u$  sera exposé ultérieurement.

La solution vient de l'écriture de la base  $\beta$  dans une base adjointe  $v$ . Cette base  $v$  admettra une base  $u$  telle que le sous espace  $\mathcal{F}(S_u^2)$  appartienne à l'espace de fonction dont  $v$  est une base. La base  $u$  est caractérisée par son ordre  $k_u$  défini par l'équation (V.52) et la condition de continuité donnée par

$$v_{u,i} = \begin{cases} 0, & \text{si } i = 1, l+1 \\ k_u - 1, & \text{si } i = 2, \dots, l \end{cases} \quad (\text{V.53})$$

Ainsi  $u$  sera la base de l'espace de fonction  $\mathbb{P}_{k_u, \xi, v_u}$ . Le sous-espace  $\mathcal{F}(\mathcal{S}_u^2)$  est alors tel que  $\mathcal{F}(\mathcal{S}_u^2) \subset \mathbb{P}_{k_v, \xi, v_v}$  avec  $k_v = k$  et

$$v_{v,i} = \begin{cases} 0 & \text{si } i = 1, l+1 \\ \frac{k-1}{2} & \text{si } k_v \text{ est impair et } i = 2, \dots, l \\ \frac{k-2}{2} & \text{si } k_v \text{ est pair et } i = 2, \dots, l \end{cases} \quad (\text{V.54})$$

On définit par la suite la base  $v$  comme la base de l'espace  $\mathbb{P}_{k_v, \xi, v_v}$ .

**Proposition V.3** Soient deux sous-espaces de polynômes par morceaux  $\mathbb{P}_{k, \xi, v}$  et  $\mathbb{P}_{k, \xi, v'}$  avec  $v_i \geq v'_i$  pour  $i = 1, \dots, l+1$ . Il vient que  $\mathbb{P}_{k, \xi, v} \subset \mathbb{P}_{k, \xi, v'}$  ■

**Démonstration**

Pour démontrer ce résultat, nous allons utiliser la base des puissances tronquées de  $\mathbb{P}_{k, \xi, v}$ . La fonction tronquée est définie par [de Boor 1978] :

$$(x - x_0)_+ = \max\{x - x_0, 0\} \quad (\text{V.55})$$

On peut alors définir la fonction puissance tronquée [de Boor 1978] :

$$(x)_+^r = (x_+)^r \quad (\text{V.56})$$

Il est démontré dans [de Boor 1978, pages 102-103] qu'une base de  $\mathbb{P}_{k, \xi, v}$  peut être obtenue à partir d'une famille libre de fonctions puissances tronquées :

$$\phi_{i,j}(x) = \begin{cases} \frac{(x - \xi_1)^j}{j!} & i = 1 \\ \frac{(x - \xi_i)_+^j}{j!} & i = 2, \dots, l \end{cases} \quad (\text{V.57})$$

avec  $j = 0, \dots, k-1$ . Ainsi  $P_\tau(x) \in \mathbb{P}_{k, \xi, v}$  peut s'écrire de façon unique dans cette base :

$$P_\tau(x) = \sum_{i=1}^l \sum_{j=v_i}^{k-1} \tau_{i,j} \phi_{i,j} \quad (\text{V.58})$$

Ainsi, la base de  $\mathbb{P}_{k, \xi, v}$  est composée des fonctions puissances tronquées  $\phi_{i,j}$ ,  $i = 1, \dots, l$  et  $j = v_i, \dots, k-1$  et la base de  $\mathbb{P}_{k, \xi, v'}$  est composée des fonctions puissances tronquées  $\phi_{i,j}$ ,  $i = 1, \dots, l$  et  $j = v'_i, \dots, k-1$ . Or par hypothèse  $v_i \geq v'_i$ , il vient alors que la base de  $\mathbb{P}_{k, \xi, v}$  est comprise dans la base de  $\mathbb{P}_{k, \xi, v'}$ . On en déduit que  $\mathbb{P}_{k, \xi, v} \subset \mathbb{P}_{k, \xi, v'}$  □

La proposition V.3 nous permet de conclure qu'une base  $\mathbb{P}_{k, \xi, v}$  peut être réécrite dans une base de  $\mathbb{P}_{k, \xi, v'}$ .

Nous allons maintenant écrire les coordonnées des éléments de la base  $\beta$  dans la base  $v$ . Chaque B-spline  $\beta_{i,k_v}$  étant de support  $\zeta(\beta_{i,k_v})$  fini, elle va se réécrire en fonction des B-splines  $v_{i,k_v}$  compatibles avec elle.

**Définition V.3**

Soient deux B-splines  $B_{i,k}$  et  $B'_{j,k}$  de même ordre issues de deux bases construites sur la même séquence  $\xi$  et de classe de continuités respectives  $v_i$  et  $v'_j$ . Nous dirons que  $B'_{j,k}$  est compatible avec  $B_{i,k}$  si :

- $\zeta(B'_{j,k}) \subseteq \zeta(B_{i,k})$ ,
- lorsque  $\underline{\zeta}(B'_{j,k}) = \underline{\zeta}(B_{i,k})$ , alors  $v'_j \geq v_i$  en  $\underline{\zeta}(B_{i,k})$
- lorsque  $\overline{\zeta}(B'_{j,k}) = \overline{\zeta}(B_{i,k})$ , alors  $v'_j \geq v_i$  en  $\overline{\zeta}(B_{i,k})$

Ainsi nous avons :

$$\beta_{i,k} = \sum_{j=\underline{\eta}_i}^{\overline{\eta}_i} \gamma_j v_{j,k} \quad (\text{V.59})$$

avec  $\underline{\eta}_i$  étant l'indice de la première B-spline de la base  $v$  compatible avec  $\beta_{i,k}$  et  $\overline{\eta}_i$  l'indice de la dernière B-spline de la base  $v$  compatible avec  $\beta_{i,k}$ . La détermination des coefficients  $\gamma_j$  utilise le théorème de Hausdorff sur le problème des moments.

**Théorème V.3 (Problème des moments)** Soient deux fonctions continues  $f, g : I \rightarrow \mathbb{R}$  dont chacune admet, pour tout entier naturel  $n$ , un moment d'ordre  $n$ . Si  $I$  est un intervalle fermé et borné, alors

$$f = g \Leftrightarrow \forall n \in \mathbb{N}, m_n(f) = m_n(g) \quad (\text{V.60})$$

■

Ainsi nous pouvons écrire pour tout  $n$  :

$$m_n(\beta_{i,k}) = m_n\left(\sum_{j=\underline{\eta}_i}^{\overline{\eta}_i} \gamma_{i,j} v_{j,k}\right) \quad (\text{V.61})$$

L'expression précédente peut se reformuler grâce à la linéarité des moments :

$$m_n(\beta_{i,k}) = \sum_{j=\underline{\eta}_i}^{\overline{\eta}_i} \gamma_{i,j} m_n(v_{j,k}) \quad (\text{V.62})$$

Le calcul des moments de B-splines a été décrit dans [Neuman 1981]. A partir de l'équation (V.62) et du calcul de  $n_\eta = \overline{\eta}_i - \underline{\eta}_i$  moments, on obtient l'équation qui permettra de calculer les coefficients  $\gamma_{i,j}$  :

$$\begin{bmatrix} m_1(v_{\underline{\eta}_i,k}) & \dots & m_1(v_{\overline{\eta}_i,k}) \\ \vdots & & \vdots \\ m_{n_\eta}(v_{\underline{\eta}_i,k}) & \dots & m_{n_\eta}(v_{\overline{\eta}_i,k}) \end{bmatrix} \begin{bmatrix} \gamma_{i,\underline{\eta}_i} \\ \vdots \\ \gamma_{i,\overline{\eta}_i} \end{bmatrix} = \begin{bmatrix} m_1(\beta_{i,k}) \\ \vdots \\ m_{n_\eta}(\beta_{i,k}) \end{bmatrix} \quad (\text{V.63})$$

Nous pouvons maintenant établir  $\Gamma$  la matrice de passage donnant  $\kappa'$ , les coordonnées de la fonctions  $\kappa(t)$  dans la base  $v$ , en fonction de  $\kappa$ , les coordonnées de  $\kappa(t)$  dans  $\beta$  (cf. (V.50)) :

$$\begin{bmatrix} \kappa_1 \\ \vdots \\ \kappa_{n_v} \end{bmatrix} = \Gamma \begin{bmatrix} \kappa'_1 \\ \vdots \\ \kappa'_{n_\beta} \end{bmatrix} \quad (\text{V.64})$$

avec la matrice  $\Gamma \in \mathbb{R}^{(n_v \times n_\beta)}$  définie par

$$\Gamma_{j,i} = \begin{cases} \gamma_{i,j} & \text{si } \underline{\eta}_i \leq j \leq \overline{\eta}_i \\ 0 & \text{autrement} \end{cases}, \quad i = 1, \dots, n_\beta, j = 1, \dots, n_v \quad (\text{V.65})$$

Le problème de positivité (V.50) va maintenant se poser dans la base  $v$  :

$$\kappa(t) = \sum_{i=1}^{n_\beta} \kappa'_i v_{i,k}(t) \geq 0 \quad (\text{V.66})$$

Le théorème V.1 (ou V.2 suivant la parité de  $k$ ) pose des conditions sur les coefficients  $\kappa'_i$  pour que la B-spline  $\kappa(t)$  soit positive ou nulle. De plus, le vecteur  $\kappa(t)$  doit aussi vérifier la relation (V.64) pour que la B-spline solution  $\kappa(t)$  appartienne à  $\mathbb{P}_{k,\xi,v}$ . Ainsi le problème à résoudre est le suivant :

$$\begin{cases} \kappa' = \Lambda^*(Y), & Y \succeq 0 \\ \kappa' = \Gamma\kappa \end{cases} \quad (\text{V.67})$$

Remarquons que les vecteurs des coefficients  $C$  et  $\kappa$  sont liés linéairement :

$$\kappa = \alpha C - b \quad (\text{V.68})$$

où

$$\alpha = [\alpha_{1,1}, \dots, \alpha_{1,n}, \alpha_{2,1}, \dots, \alpha_{2,n}, \alpha_{3,1}, \dots, \alpha_{3,n}] \quad (\text{V.69})$$

et

$$C = [C_{1,1}, \dots, C_{1,n}, C_{2,1}, \dots, C_{2,n}, C_{3,1}, \dots, C_{3,n}]^T \quad (\text{V.70})$$

Nous pouvons maintenant décrire la condition d'appartenance d'une trajectoire  $t \mapsto \bar{z}(t)$  à un demi-espace de  $O_{\bar{z}}$

**Théorème V.4** Les coefficients  $C$  répondent au problème (V.46) si et seulement si il existe une matrice  $Y$  semi-définie positive telle que :

$$\Gamma(\alpha C - b) = \Lambda^*(Y), \quad Y \succeq 0 \quad (\text{V.71})$$

■

Si  $r \geq k_u$  alors il existe la base  $u$  telle que  $\beta$  soit équivalente à la base de représentation adjointe  $v$ . Les théorèmes peuvent alors être exploités de façon plus directe. Le théorème précédent devient alors :

**Théorème V.5** Les coefficients  $C$  répondent au problème (V.46) si et seulement si il existe une matrice  $Y$  semi-définie positive telle que :

$$\alpha C - b = \Lambda^*(Y), \quad Y \succeq 0 \quad (\text{V.72})$$

■

La méthode développée dans ce paragraphe peut se résumer dans la démarche suivante :

1. Etablir la base  $B$  décrivant les trajectoires et recenser le nombre de dérivées en jeu dans la contrainte (V.42).
2. Etablir la base  $\beta$  ainsi que la relation linéaire (V.51).
3. Calculer l'ordre  $k_u$  de la base  $u$  :

Si  $r \geq k_u$  alors la base  $\beta$  équivaut à la base  $v$  et les coefficients  $C$  répondant à (V.42) sont donnés par le théorème V.5.

Si  $r < k_u$ , alors il faut établir une base  $v$  telle que décrite ci-dessus et utiliser le théorème V.4 pour déterminer les coefficients  $C$  répondant à (V.42)

Nous allons, dans le paragraphe suivant, illustrer les différentes étapes sur un exemple académique.

### V.3.1.3 Exemple

Considérons un système non linéaire composé d'un état et piloté par une entrée et défini par :

$$\dot{x} = -ax^2 + u \quad (\text{V.73})$$

Ce système est plat et admet la sortie plate  $z$  telle que :

$$z = x \quad (\text{V.74})$$

La trajectoire de la sortie plate est soumise aux contraintes transverses suivantes :

$$\begin{cases} x_{t_0} = 0 & \dot{x}_{t_0} = 0 \\ x_{t_f} = 0.5 & \dot{x}_{t_f} = 0.2 \end{cases} \quad (\text{V.75})$$

Les contraintes sur la trajectoire du système (V.73) imposent :

$$\dot{z} - z \leq 1 \quad (\text{V.76})$$

1. Nous choisissons de paramétrer la trajectoire  $z(t)$  dans la base  $B$  de B-splines d'ordre 7 sur l'intervalle  $\Delta = [0, 10]$  et définie sur  $l = 10$  segments telle que :

$$\begin{aligned} T_B &= \{t_1, \dots, t_B\} \\ &= \underbrace{\{\xi_1, \dots, \xi_1\}}_{\rho_{B,1} \text{ fois}} \underbrace{\{\xi_2, \dots, \xi_2\}}_{\rho_{B,2} \text{ fois}} \dots \underbrace{\{\xi_{l+1}, \dots, \xi_{l+1}\}}_{\rho_{B,l+1} \text{ fois}} \end{aligned} \quad (\text{V.77})$$

avec les séquences  $\xi$  et  $\rho_B$  définies par :

$$\begin{aligned} \xi_i &= i - 1 \\ \rho_{B,i} &= \begin{cases} 7 & \text{si } i = 1, l+1, \quad i = 1, \dots, l+1 \\ 1 & \text{si } i = 2, \dots, l \end{cases} \end{aligned} \quad (\text{V.78})$$

La base  $B$  est de dimension 16.

2. Le nombre  $r$  de dérivées est de 1. La base  $\beta$  est alors la base d'ordre 7 construite sur la séquence  $T$  suivante :

$$\begin{aligned} T_\beta &= \{t_1, \dots, t_\beta\} \\ &= \underbrace{\{\xi_1, \dots, \xi_1\}}_{\rho_{\beta,1} \text{ fois}} \underbrace{\{\xi_2, \dots, \xi_2\}}_{\rho_{\beta,2} \text{ fois}} \dots \underbrace{\{\xi_{l+1}, \dots, \xi_{l+1}\}}_{\rho_{l+1} \text{ fois}} \end{aligned} \quad (\text{V.79})$$

avec les séquences  $\xi$  et  $\rho_\beta$  définies par :

$$\begin{aligned} \xi_i &= i - 1 \\ \rho_{\beta,i} &= \begin{cases} 7 & \text{si } i = 1, l+1, \quad i = 1, \dots, l+1 \\ 2 & \text{si } i = 2, \dots, l \end{cases} \end{aligned} \quad (\text{V.80})$$

La matrice  $\alpha$  est calculée à partir des méthodes présentées dans le paragraphe précédent.

3.  $k$  étant impair, nous avons  $k_u = \frac{k+1}{2} = 4$ . Il vient que  $r < k_u$ . Ainsi, d'après la méthode présentée dans le paragraphe précédent, la base  $\nu$  d'ordre 7 est définie par :

$$\begin{aligned} T_\nu &= \{t_1, \dots, t_\nu\} \\ &= \underbrace{\{\xi_1, \dots, \xi_1\}}_{\rho_1 \text{ fois}} \underbrace{\{\xi_2, \dots, \xi_2\}}_{\rho_2 \text{ fois}} \dots \underbrace{\{\xi_{l+1}, \dots, \xi_{l+1}\}}_{\rho_{l+1} \text{ fois}} \end{aligned} \quad (\text{V.81})$$

avec les séquences  $\xi$  et  $\rho_\nu$  définies par :

$$\begin{aligned} \xi_i &= i - 1 \\ \rho_{\nu,i} &= \begin{cases} 7 & \text{si } i = 1, l+1, \quad i = 1, \dots, l+1 \\ 4 & \text{si } i = 2, \dots, l \end{cases} \end{aligned} \quad (\text{V.82})$$

La base  $\nu$  comporte 43 éléments. La matrice de passage  $\Gamma$  est alors construite en utilisant les équations (V.63) et (V.65). La base  $u$  associée à  $\nu$  est d'ordre 4 et est définie sur la séquence  $T_u$  suivante :

$$\begin{aligned} T_u &= \{t_1, \dots, t_u\} \\ &= \underbrace{\{\xi_1, \dots, \xi_1\}}_{\rho_1 \text{ fois}} \underbrace{\{\xi_2, \dots, \xi_2\}}_{\rho_2 \text{ fois}} \dots \underbrace{\{\xi_{l+1}, \dots, \xi_{l+1}\}}_{\rho_{l+1} \text{ fois}} \end{aligned} \quad (\text{V.83})$$

avec les séquences  $\xi$  et  $\rho_u$  définies par :

$$\begin{aligned} \xi_i &= i - 1 \\ \rho_{u,i} &= \begin{cases} 4 & \text{si } i = 1, l+1, \quad i = 1, \dots, l+1 \\ 1 & \text{si } i = 2, \dots, l \end{cases} \end{aligned} \quad (\text{V.84})$$

La dimension de la base  $u$  est 13.

Le problème (V.71) est résolu grâce au solveur SDPT3 [Toh & Tutuncu 1999] et conduit à un résultat en 0.25 secondes. La trajectoire obtenue est présentée sur la figure V.1.

Reprenons maintenant cet exemple dans le cas où la trajectoire de la sortie plate est paramétrée dans une base de B-splines  $B$  d'ordre pair  $k = 8$ . La base  $B$  est définie sur la séquence  $\xi$  et les multiplicités associés  $\rho_B$  :

$$\begin{aligned} \xi_i &= i - 1 \\ \rho_{B,i} &= \begin{cases} 8 & \text{si } i = 1, l + 1, \quad i = 1, \dots, l + 1 \\ 1 & \text{si } i = 2, \dots, l \end{cases} \end{aligned} \quad (\text{V.85})$$

Avec  $r = 1$ , la base  $\beta$  du problème se définit sur la base  $\xi$  et  $\rho_\beta$

$$\begin{aligned} \xi_i &= i - 1 \\ \rho_{\beta,i} &= \begin{cases} 8 & \text{si } i = 1, l + 1, \quad i = 1, \dots, l + 1 \\ 2 & \text{si } i = 2, \dots, l \end{cases} \end{aligned} \quad (\text{V.86})$$

$k$  étant pair,  $k_u = \frac{k}{2} = 4$ . Nous définissons la base  $u$  d'ordre  $k_u = 4$  sur les séquence  $\xi$  et  $\rho_u$  telle que :

$$\begin{aligned} \xi_i &= i - 1 \\ \rho_{u,i} &= \begin{cases} 4 & \text{si } i = 1, l + 1, \quad i = 1, \dots, l + 1 \\ 1 & \text{si } i = 2, \dots, l \end{cases} \end{aligned} \quad (\text{V.87})$$

Afin d'obtenir la base  $v$  et l'opérateur  $\Lambda(v)$ , nous définissons la fonction  $\bar{\tau}$  définie dans le paragraphe V.2.2. telle que :

$$\bar{\tau}(t) = t + 1$$

La base  $v$  d'ordre 8 est alors définie sur les séquence  $\xi$  et  $\rho_v$  :

$$\begin{aligned} \xi_i &= i - 1 \\ \rho_{v,i} &= \begin{cases} 8 & \text{si } i = 1, l + 1, \quad i = 1, \dots, l + 1 \\ 5 & \text{si } i = 2, \dots, l \end{cases} \end{aligned} \quad (\text{V.88})$$

Elle comporte 53 éléments. Remarquons que  $k_u \geq r$ , nous utiliserons ainsi le théorème V.4. La matrice de passage  $\Gamma$  se calcule de la même manière que précédemment. Le résultat de l'application numérique est présenté sur la figure V.1.

Nous allons maintenant aborder le problème d'appartenance d'une trajectoire à l'intersection de demi-espaces délimités par des hyperplans.

### V.3.2 Appartenance d'une trajectoire $t \mapsto \bar{z}(t)$ à l'intersection de plusieurs demi-espaces de $O_{\bar{z}}$

#### V.3.2.1 Formulation du problème

Considérons  $n_c$  hyperplans de l'espace  $O_{\bar{z}}$ . Les trajectoires des sorties plates sont paramétrées par des B-splines (cf. (III.15)). Le problème d'appartenance d'une trajectoire  $t \mapsto \bar{z}(t)$  à un demi

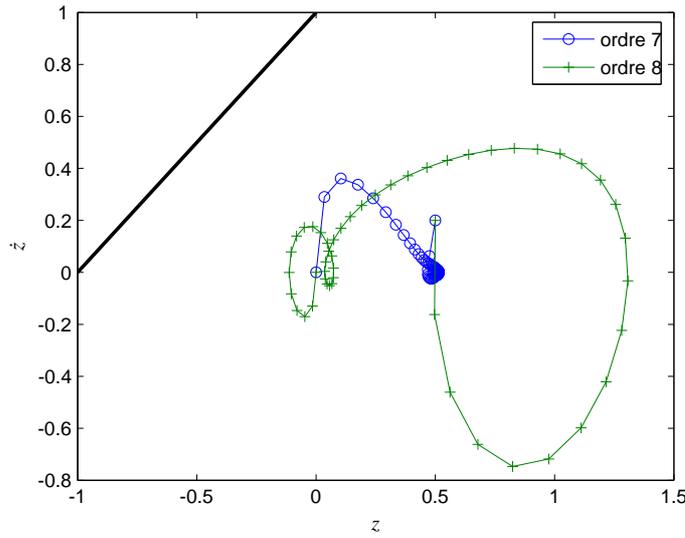


FIG. V.1 – Trajectoire pour une base de B-spline d'ordre 7 et 8 sous contrainte linéaire

espace ouvert se définit comme un problème de détermination d'une matrice semi-définie positive  $Y$  telle que

$$\Gamma(\alpha C - b) = \Lambda^*(Y), \quad Y \succeq 0 \quad (\text{V.89})$$

où  $\alpha$  et  $b$  sont caractéristiques de l'hyperplan. L'appartenance d'une trajectoire  $t \mapsto \bar{z}(t)$  à l'intersection de plusieurs demi-espaces de  $O_{\bar{z}}$  va se traduire par la conjonction des différents problèmes attenants aux différents hyperplans :

$$\begin{cases} \Gamma(\alpha_1 C_1 - b_1) = \Lambda^*(Y_1), & Y_1 \succeq 0 \\ \vdots \\ \Gamma(\alpha_{n_c} C_{n_c} - b_{n_c}) = \Lambda^*(Y_{n_c}), & Y_{n_c} \succeq 0 \end{cases} \quad (\text{V.90})$$

Afin de rendre le problème cohérent au regard du problème de génération de trajectoire dans lequel nous désirons un vecteur  $C$  unique, nous ajoutons une contrainte supplémentaire :

$$\begin{cases} C = C_1 = \dots = C_{n_c} \\ \Gamma(\alpha_1 C_1 - b_1) = \Lambda^*(Y_1), & Y_1 \succeq 0 \\ \vdots \\ \Gamma(\alpha_{n_c} C_{n_c} - b_{n_c}) = \Lambda^*(Y_{n_c}), & Y_{n_c} \succeq 0 \end{cases} \quad (\text{V.91})$$

**Remarque V.3** L'adjonction de cette dernière contrainte dégrade la condition nécessaire et suffisante du système (V.90) en condition uniquement suffisante.

La prise en compte des contraintes transverses (conditions initiales et finales) se traduit par une condition linéaire sur le vecteur des coefficients des B-splines :

$$M_c C = M_z \quad (\text{V.92})$$

où la matrice de collocation  $M_c$  évalue la base des B-splines et de leurs dérivées successives aux instants  $t_0$  et  $t_f$ , et  $M_z$  évalue le vecteur  $\bar{z}$  aux mêmes instants. Le problème (V.91) augmenté

des contraintes transverses s'écrit alors :

$$\begin{cases} M_c C = M_z \\ \Gamma(\alpha_1 C - b_1) = \Lambda^*(Y_1), & Y_1 \succeq 0 \\ \vdots \\ \Gamma(\alpha_{n_c} C - b_{n_c}) = \Lambda^*(Y_{n_c}), & Y_{n_c} \succeq 0 \end{cases} \quad (\text{V.93})$$

**Théorème V.6** Une condition suffisante pour que la trajectoire  $t \mapsto \bar{z}(t)$  définie par (III.15) appartienne à l'intersection de  $n_c$  demi-espaces est qu'il existe  $n_c$  matrices  $Y_i$  semi-définies positives telles que la condition (V.93) soit vérifiée. ■

### V.3.2.2 Exemple

Reprenons l'exemple défini dans le paragraphe V.3.1.3. Nous désirons maintenant que la trajectoire  $t \mapsto \bar{z}(t)$  soit inscrite dans le polytope du plan  $O_{\bar{z}}$  définie par l'inégalité matricielle suivante :

$$\begin{bmatrix} -1 & -1 \\ -1 & 1 \\ 1 & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} z \\ \dot{z} \end{bmatrix} \leq \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} \quad (\text{V.94})$$

La trajectoire  $t \mapsto \bar{z}(t)$  est définie dans la base  $B$  construite sur  $T_B$  (V.77). On en déduit que les bases  $\beta$ ,  $v$ ,  $u$  sont les mêmes que celles établies dans le paragraphe V.3.1.3 respectivement pour un ordre 7 et 8. Nous construisons alors un problème équivalent au problème (V.93) que nous résolvons grâce à SDPT3 [Toh & Tutuncu 1999]

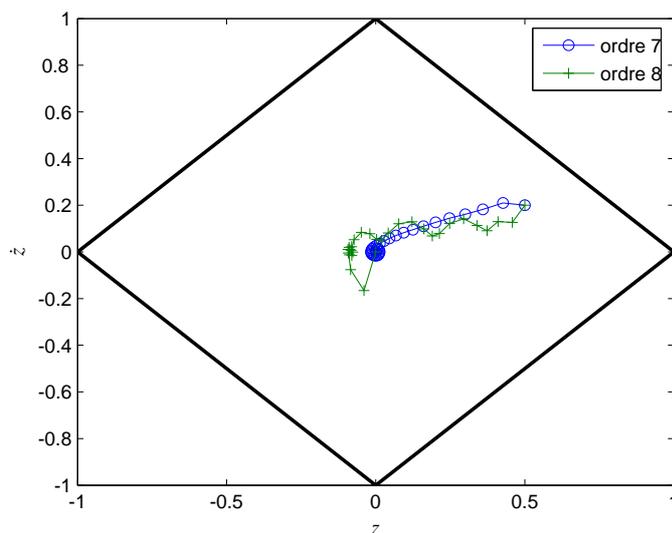


FIG. V.2 – Trajectoire pour une base de B-spline d'ordre 7 et 8 sous contrainte d'appartenance à un polytope

### V.3.3 Application au problème de guidage

Le problème de guidage que nous souhaitons résoudre est le suivant :

$$\begin{aligned} & \min_C J_C(C) \\ \text{sous les contraintes : } & \begin{cases} \bar{z}(t_i, C) = \bar{z}_i \\ \bar{z}(t_f, C) = \bar{z}_f \\ \bar{z}(C) \in S_I \end{cases} \end{aligned} \quad (\text{V.95})$$

La région admissible  $S_I$  impliquée dans le problème que nous considérons est définie par la surface  $S_6$  qui décrit un polytope inscrit dans l'espace admissible  $S_{\bar{z}}$ . Notons  $H$  et  $K$  les matrices de la représentation par hyperplan de  $S_6$  telles que :

$$\forall t \in \Delta, H\bar{z}(t) \leq K \Leftrightarrow \bar{z}(t) \in S_6 \quad (\text{V.96})$$

La surface  $S_6$  est un polytope défini par 512 hyperplans. Malgré l'usage de différents solveurs, nous avons rencontré d'importants problèmes numériques. En considérant des trajectoires de sorties plates définies sur la base  $B$  du paragraphe V.3.1.3, le problème posé ici est différent de celui posé dans le paragraphe V.3.2.2, essentiellement à cause du nombre de demi-espaces considérés du point de vue du solveur. Il peut alors être intéressant de réfléchir à des méthodes de réduction de dimension du problème  $\mathcal{LMI}$  développé ici.

Considérons  $S'_6$ , la projection de  $S_6$  dans l'espace  $O'_z$  de coordonnées  $\bar{z}' = \{\dot{z}_1, \dot{z}_2, \dot{z}_3, \ddot{z}_1, \ddot{z}_2, \ddot{z}_3\}$ . Nous posons le problème de commande optimale réduit suivant :

$$\begin{aligned} & \min_C J_C(C) \\ \text{sous les contraintes : } & \begin{cases} \bar{z}(t_i, C) = \bar{z}_i \\ \bar{z}(t_f, C) = \bar{z}_f \\ \bar{z}'(C) \in S'_6 \end{cases} \end{aligned} \quad (\text{V.97})$$

**Proposition V.4** *Une trajectoire bornée  $t \mapsto \bar{z}(t)$  solution au problème (V.97) est une solution au problème de guidage.* ■

#### Démonstration

En effet une condition nécessaire pour qu'une trajectoire  $t \mapsto \bar{z}(t)$  appartienne à  $S_6$  est que la trajectoire  $t \mapsto \bar{z}'(t)$  appartienne à  $S'_6$ . De plus, la remarque III.1 de la page 53 nous indique que le rayon de la sphère dans laquelle doit être inscrite la trajectoire  $t \mapsto (z_1(t), z_2(t), z_3(t))$  est extensible. En d'autres termes, cette contrainte signifie que la trajectoire  $t \mapsto (z_1(t), z_2(t), z_3(t))$  doit être bornée. □

La résolution du problème (V.97) débute par la définition d'une base  $B^*$  d'ordre  $k_{B^*}$  et de degré de continuité  $v_{B^*}$ . Les sorties plates  $z_1, z_2, z_3$  sont décrites dans cette base  $B^*$ . Ainsi les dérivées premières  $\dot{z}_1, \dot{z}_2, \dot{z}_3$  peuvent être décrites dans une base  $B$  d'ordre  $k_B = k_{B^*} - 1$  et de degré de continuité  $v_B = v_{B^*} - 1$ . On peut alors résoudre le problème (V.97) par la mise en oeuvre de la démarche du paragraphe V.3.1.2 en considérant que le nombre de dérivées est égale à 1. Ensuite l'application du théorème V.6 nous amène à résoudre le problème (V.93) où les coefficients  $C$  définissent la trajectoire  $t \mapsto (\dot{z}_1, \dot{z}_2, \dot{z}_3)$  dans la base  $B$ . De plus, il existe une

matrice de passage  $\Gamma^*$  entre les coefficients  $C^*$  définissant la trajectoire  $t \mapsto (z_1(t), z_2(t), z_3(t))$  dans la base  $B^*$  et les coefficients  $C$  tels que :

$$C = \Gamma^* C^*$$

Ainsi le problème d'optimisation  $\mathcal{LMI}$  à résoudre est le suivant :

$$\begin{cases} M_c C^* = M_z \\ \alpha_1 \Gamma^* C^* - b_1 = \Lambda^*(Y_1), & Y_1 \succeq 0 \\ \vdots \\ \alpha_{n_c} \Gamma^* C^* - b_{n_c} = \Lambda^*(Y_{n_c}), & Y_{n_c} \succeq 0 \end{cases} \quad (\text{V.98})$$

Une première solution fournie par la résolution de (V.98) correspond à une durée de la manoeuvre de 50 secondes. Cette durée est comparable à celle obtenue avec l'approche du chapitre III en considérant des contraintes linéaire (cf. paragraphe III.5.4.2). Notons, enfin, que cette durée correspond à une première itération qu'un processus de recherche du temps optimal comme proposé à la figure III.12 permettrait d'optimiser. La durée de la manoeuvre est de 50 secondes. La base  $B^*$  est d'ordre 8 sur 10 segments uniformes et de continuité maximale aux noeuds de segmentation. La durée d'exécution du problème (V.98) par le solveur SDPT3 [Toh & Tutuncu 1999] est approximativement de 6.3 secondes. La trajectoire obtenue est présentée sur la figure V.3.

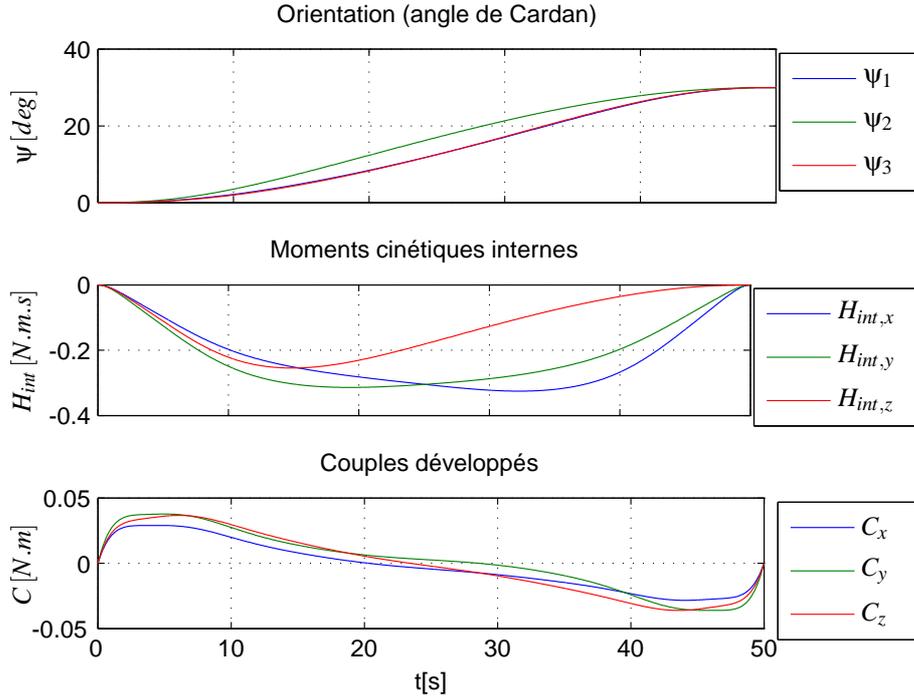


FIG. V.3 – Trajectoire pour une base de B-spline d'ordre 8 sous contrainte linéaire

## V.4 Conclusions

Ce chapitre avait comme objectif de poser les bases méthodologiques d'une nouvelle approche permettant de vérifier les contraintes imposées sur le continuum temporel et non plus

---

en un nombre fini d'instants. Nous avons formulé le problème de génération de trajectoires des sorties plates  $t \mapsto \bar{z}(t)$  sous contraintes linéaires comme la conjonction de problèmes de positivité de polynômes par morceaux. Nous avons montré que chaque problème de positivité se définit en termes de positivité semi-définie d'une matrice. Par le biais d'un exemple, nous avons montré que les outils de la programmation semi-définie permettent d'obtenir des trajectoires qui vérifient les contraintes de façon continue dans le temps. Enfin, nous avons montré comment la méthodologie développée dans ce chapitre pouvait s'appliquer au problème de guidage



# Conclusion générale

L'objectif général de cette thèse était de développer des méthodes et des outils de génération de trajectoires d'attitude pour satellites rigides pilotés par des roues à réaction.

Le chapitre I avait pour objectif de préciser les hypothèses de travail en se basant sur une analyse critique des méthodes existantes. Nous avons tout d'abord présenté un tour d'horizon des différentes méthodes de génération de trajectoires classiquement utilisées dans la littérature. Nous nous sommes efforcés de présenter les principales techniques, de clarifier leurs périmètres d'application potentielle au problème de guidage et d'identifier les problèmes encore ouverts. Ce tour d'horizon nous a permis de motiver et de justifier les orientations méthodologiques choisies dans la suite de ce mémoire. Nous avons en particulier souligné le fait que pour les supports applicatifs cibles (les plateformes de type Myriade ou Proteus), pour lesquels les actionneurs sont des roues à réaction, il est nécessaire de développer des méthodes de génération de trajectoires produisant des profils de couples adaptés (suffisamment continus).

Au cours du deuxième chapitre, nous avons tout d'abord démontré le caractère différentiellement plat du modèle dynamique d'attitude d'un satellite rigide et déterminé un jeu de sorties plates. Nous avons ensuite mis en place un problème d'optimisation ne nécessitant pas d'intégration en exploitant l'impact de la platitude différentielle sur le problème de commande optimale. Dans les chapitres suivants, deux méthodes distinctes de résolution de ce problème d'optimisation *géométrique* sont abordées.

Dans le chapitre III, nous avons associé les méthodes directes de collocation du problème de commande optimale à la platitude différentielle. Nous avons repris et spécifié le problème de génération de trajectoires de sorties plates sous contraintes définies dans le chapitre II. Le paramétrage des trajectoires des sorties plates par des B-splines nous a permis de formuler le problème en termes de programmation semi-infinie. Nous en avons déduit alors le problème d'optimisation paramétrique non linéaire par discrétisation du problème de satisfaction de contraintes. Ce problème d'optimisation non convexe est alors résolu par des algorithmes de programmation non linéaire. Afin de simplifier cette résolution numérique, la région admissible définie par les contraintes sur les sorties plates a été approximée par des volumes convexes (superellipsoïde) et intérieurs à celle-ci. Une étude numérique approfondie a permis d'examiner les avantages et les inconvénients de l'utilisation de ce schéma convexe de génération de trajectoires. Au cours de cette étude numérique, nous avons mis en évidence la nécessité d'effectuer une analyse *a posteriori* afin de vérifier si les contraintes sont bien satisfaites entre les points de collocation. Ce constat, intrinsèque à la notion de collocation, est à l'origine des développements présentés dans le chapitre V.

Le chapitre IV a été consacré aux résultats obtenus sur un simulateur interne au CNES. Ce chapitre a mis en évidence l'intérêt de la platitude par rapport aux techniques développées par le CNES et qui s'appuient sur des méthodes classiques pour générer des profils de guidage. Ces

premiers résultats sont encourageants et laissent à penser que la méthodologie développée au chapitre III, associée à une procédure de post-analyse, pourrait constituer un outil opérationnel sous réserve d'une validation par des tests supplémentaires (robustesse du code informatique, scénarios de manoeuvre, ergonomie de l'outil...). Une analyse pertinente des paramètres de collocation (notamment la distribution des points de segmentation et de collocation) pourrait permettre d'alléger le processus de vérification a posteriori.

Enfin, le chapitre V a été consacré au développement d'une contribution majeure de ce mémoire. Nous avons posé un cadre méthodologique visant à prendre en compte les contraintes de façon continue dans le temps. Pour ce faire, nous avons considéré un problème de génération de trajectoires de sorties plates sous contraintes linéaires. Nous avons montré qu'un tel problème pouvait s'écrire en termes de plusieurs problèmes de positivité de polynômes par morceaux. Nous avons de plus démontré que la positivité d'un polynôme par morceaux peut s'exprimer comme un lien entre le cône des coefficients dans la base de B-splines associée à ce polynôme et le cône des matrices semi-définies positives. Ainsi, nous avons ouvert la possibilité de résoudre un problème de génération de trajectoires d'un système plat non linéaire sous contraintes par des outils de programmation semi-définie. Nous avons ensuite illustré la démarche globale sur un exemple académique, avant d'appliquer la méthodologie au problème de guidage en attitude.

Les résultats obtenus dans ce mémoire offrent des perspectives intéressantes de développements ultérieurs :

En premier lieu et au plan méthodologique : en amont du problème de génération de trajectoires, se pose la question de l'existence d'un critère calculable et systématique qui permette de conclure à la platitude, et le cas échéant, de caractériser les sorties plates candidates. Cette question n'a pas encore fait l'objet d'une résolution satisfaisante. En effet, si plusieurs conditions nécessaires et suffisantes, dont celle récemment démontrée dans [Lévine 2004*b*], existent désormais, de véritables techniques systématiques et algorithmiques de détermination des sorties plates n'ont toujours pas été dégagées.

Ensuite au plan applicatif, il faudra sans doute plusieurs campagnes de test sur les simulateurs industriels pour valider la méthodologie développée au chapitre V. En effet, pour l'instant, ces techniques basées sur la positivité sont uniquement testées par des simulations informatiques. Malgré les résultats encourageants obtenus, la démarche nécessite certainement d'être confrontée à différents scénarios de guidage. De plus, la dimension élevée du problème LMI qui en résulte peut être un frein à son utilisation. Des études supplémentaires seront nécessaires pour rendre la méthode plus robuste en grande dimension, en gérant intelligemment les compromis de synthèse pour le problème de guidage considéré.

# Références bibliographiques

## Conférences internationales à comité de lecture

- Louembet, C., Cazaurang, F., Zolghadri, A., Pittet, C. & Maini, M. [2006a], Flatness and collocation based methodology for satellite trajectory planning, *dans* 'IEEE International Symposium on Control, Communications, and Signal Processing', Marrakech, Morocco.
- Louembet, C., Cazaurang, F., Zolghadri, A., Pittet, C. & Maini, M. [2006], Génération de trajectoire pour satellite agile par platitude et collocation, *dans* 'Proceedings of Conférence Internationale Francophone d'Automatique CIFA 2006, Bordeaux'.
- Louembet, C., Cazaurang, F., Zolghadri, A., Pittet, C. & Charbonnel, C. [2007], A convex approach to the design of optimal trajectories for satellite, *dans* '17<sup>th</sup> IFAC Symposium on Automatic Control in Aerospace', Toulouse, France.
- Louembet, C., Cazaurang, F., Zolghadri, A., Pittet, C. & Charbonnel, C. [2007], Design of algorithms for satellite slew manoeuvre by flatness and collocation, *in* '26<sup>th</sup> American Control Conference (ACC)', New York City, USA.



## Références bibliographiques

- Agrawal, S., Claewplodtook, P. & Fabien, B. [1998], 'Optimal trajectories of open-chain robot systems : a newsolution procedure without lagrange multipliers', *ASME Journal of Dynamic Systems, Measurement and Control* . 52
- Agrawal, S. & Faiz, N. [1998], 'A new efficient method for optimization of nonlinear dynamic systems without lagrange multipliers', *Journal of Optimization Theory and Applications* . 52
- Anderson, R. & Ibragimov, N. [1979], *Lie-Bäcklund Transformation in Applications*, SIAM, Philadelphia, PA. 27
- Aranda-Bricaire, E., Moog, C. & Pomet, J.-B. [1994], Infinitesimal brunovsky form for nonlinear systems with applications to dynamic linearization, Rapport de recherche, INRIA. 26
- Ascher, U., Christiansen, J. & Russell, R. [1979], Colsys – a collocation code for boundary-value problems, in B. Childs, J. Scott, M. Daniel, E. Denman & N. P., eds, 'Codes for boundary value problems in ordinary differential Equations', Vol. 76, Springer-Verlag, Berlin. 7, 18
- Ascher, U., Mattheij, R. & R., R. [1988], *Numerical solution of boundary value problems in ordinary differential equations*, Prentice-Hall, Englewoods Cliffs NJ. 12
- Ascher, U., Press, S. & Russell, R. [1983], 'On spline basis selection for solving differential equations', *SIAM Journal of Numerical Analysis* **20**(1), 121–142. 54
- Balkanyi, L. & Spurlock, O. [1993], Duksup – a high thrust trajectory code, in 'Proceedings of the AIAA/AHS/ASSEE Aerospace Design Conference', AIAA, Washington DC. 7
- Barr, A. [1981], 'Superquadrics and angle-preserving transformations', *IEEE Computer Graphics and Applications* **1**, 11–22. 59, 60
- Betts, J. [1977], 'Optimal three-burn orbital transfer', *J. Guidance, Control and Dynamics* **15**, 861–864. 7, 20
- Betts, J. T. [1998], 'Survey of numerical methods for trajectory optimization', *J. Guidance, Control and Dynamics* **21**(2), 193–207. 7, 20
- Bilimoria, K. & Wie, B. [1993], 'Time-optimal three-axis reorientation of a rigid spacecraft', *AIAA Journal of Guidance, Control, and Dynamics* **16**(3), 446–452. 7, 16, 17
- Bliek, C. [1992], Computer Methods for Design Automation, PhD thesis, Massachussets Institute of Technology. 68
- Bock, H. & Plitt, K. [1984], A multiple shooting algorithm for direct solution of optimal control problems, in 'IFAC 9<sup>th</sup> World congress', Budapest, Hongrie. 7, 18
- Boissonnat, J. & Yvinec, M. [1995], *Géométrie Algorithmique*, Ediscience Int. 65

- Boyd, S., El Ghaoui, L., Feron, E. & Balakrishnan, V. [1994], *Linear Matrix Inequalities in System and Control Theory*, SIAM, Philadelphia. 57, 61, 64
- Brauer, G., Cornick, D. & Stevenson, R. [1977], Capabilities and applications of the program to optimize simulated trajectories (post), Technical Report CR-2270, NASA. 20
- Brenan, K., Campbell, S. & Petzold, L. [1989], *Numerical solution of initial-value problems in Differential-Algebraic equations*, North-Holland, New York. 12
- Brenan, K. & Hallman, W. [1994], Progress in systems and control theory, in J. Borggaard, J. Burkardt, M. Gunzburger & J. Peterson, eds, 'Optimal Design and Control', Vol. 19, Birkhäuser Verlag, pp. 117–132. 22
- Brusch, R. [1979], 'Constrained impulsive trajectory optimization for orbit-to orbit transfer', *J. Guidance, Control and Dynamics* **2**, 204–212. 20
- Bryson, A. [1999], *Dynamic Optimization*, Addison Wesley. 6
- Bryson, A. & Ho, Y. [1975], *Applied Optimal Control*, Wiley, New York. 13, 17
- Bulirsch, R. [1971], Die mehrzielmethode zur numerischen lösung von nichtlinearen randwertproblemen und aufgaben der optimalen steuerung, Technical report, Carl-Cranz-Gesellschaft, Oberpfaffenhofen, Germany. 7, 16
- Bulirsch, R. & Chudej, K. [1991], Ascent optimization of an airbreathing space vehicle, in 'Guidance, Navigation and Control Conference', AIAA, New Orleans. 7, 17
- Bulirsch, R., Nerz, E., Pesch, H. & von Stryk, O. [1993], *Optimal Control*, Birkhäuser Verlag, chapter Combining direct and indirect methods in nonlinear optimal control : Range maximization of a hang glider in *Optimal Control*, pp. 273–288. 17, 23
- Butcher, J. [1964], 'Implicite runge-kutta methods', *Mathematics of computation* **18**(85), 50–64. 20
- Byers, R. & Vadali, S. [1993], 'Quasi-closed-form solution to the time-optimal rigid spacecraft reorientation problem', *AIAA Journal of Guidance, Control, and Dynamics* **16**(3), 453–461. 23
- Chowdhry, R. & Cliff, E. [1991], 'Optimal rigid body motions, pat 2 : Minimum time solutions', *Journal of Optimization Theory and Applications* **70**(2). 17, 23
- Cohn, P. [1985], *Free Rings and Their Relations*, Academic Press, London. 35
- Curry, H. & Schoenberg, I. [1966], 'On pólya frequency functions : The fundamental spline functions and their limits', *J. d'Analyse Math.* **17**, 71–107. 98, 103
- d'Andrea Novel, B., Campion, G. & Bastin, G. [1995], 'Control of wheeled mobile robots not satisfying ideal velocity constraints', *Int. J. of Robust and Nonlinear Control* pp. 243–267. 40
- de Boor, C. [1978], A practical guide to spline, in 'Applied Mathematical Sciences', Vol. 27, Springer. 54, 105, 136, 137
- de Boor, C. & Fix, G. [1973], 'Spline approximation by quasi-interpolants', *Journal of Approximation Theory* **8**, 19–45. 137
- Dickmanns, E. & Well, H. [1975], Approximate solution of optimal control problems using third-order hermite polynomial functions, in Springer-Verlag, ed., 'Proceedings of the 6<sup>th</sup> Technical Conference on Optimization Techniques'. 20
- Dulot, J. [2003], Repère et cinématique d'attitude des satellites, Technical report, CNES. 8, 70

- Edelbaum, T., Sackett, L. & Malchow, H. [1973], Optimal low thrust geocentric transfer, *in* 'Proceedings of the 10<sup>th</sup> Electric Propulsion Conference', AIAA, New York. 7
- Faiz, N. [1999], Real-Time and Optimal trajectory generation for nonlinear systems, PhD thesis, University of Delaware. 57, 58
- Faiz, N. & Agrawal, S. [1998], Optimal planning of an underactuated planar body using higher-order method, *in* 'Proceedings of the IEEE International Conference on robotics and Automation'. 52
- Faiz, N., Agrawal, S. & Murray, R. [2001], 'Differentially flat systems with inequality constraints : An approach to real-time feasible trajectory generation', *J. Guidance, Control and Dynamics* **24**. 57, 58, 59
- Ferreira, A. [2001], Aspects of Flatness Based Optimal Planning and Control of Dynamic Systems, PhD thesis, Faculty of the University of Delaware. 52
- Fliasher, R., Mehlor, K., Rote, G., Welzl, E. & Yap, C. [1992], 'Simultaneous inner and outer approximation of shapes', *Algorithmica* **8**, 365–389. 57
- Fliess, M., Lévine, J., Martin, P. & Rouchon, P. [1992a], 'On differentially flat nonlinear systems', *Proc. 2nd. IFAC NOLCOS Symposium Bordeaux* . 26, 32
- Fliess, M., Lévine, J., Martin, P. & Rouchon, P. [1992b], 'Sur les systèmes non linéaires différentiellement plats', *C.R. Acad. Sci. Paris* . 26, 32, 33
- Fliess, M., Lévine, J., Martin, P. & Rouchon, P. [1995], 'Flatness and defect of non linear systems : introductory theory and examples', *Int. J. Control* **61**(6), 1327–1361. 51
- Fliess, M., Lévine, J., Martin, P. & Rouchon, P. [1999], 'a lie-bäcklund approach to equivalence and flatness of nonlinear systems', *IEEE Trans. Automat. Contr.* **44**(5), 922–937. 26, 31, 33, 51
- Fossas, E., Franch, J. & Palau, A. [1998], 'Flatness, tangent systems and flat outputs', *Proceedings Of the American Control Conference* . 26
- Gil, P., Murray, W. & Wright, M. [1981], *Practical Optimization*, Addison-Wesley. 129
- Granvilliers, L. [2004], *RealPaver User's Manual*, 0.4 edn. 66, 68
- Granvilliers, L. & Benhamou, F. [2007], 'REALPAVER : An interval solver using constraint satisfaction techniques', *ACM Transactions on Mathematical Software* . 145
- Hansen, E. [1992], 'Bounding the solution of interval linear equations', *SIAM Journal on Numerical Analysis* **29**(5), 1493–1503. 146
- Hargraves, C. & Paris, S. [1987], 'Direct trajectory optimization using nonlinear programming and collocation', *AIAA J. Guidance and Control* **10**. 7, 18, 54
- Hargreaves, G. [2002], *Interval Analysis in Matlab*, Dpt of Mathematics, University of Manchester. 68
- Henrion, D. & Lasserre, J. B. [2003], Lmis for constrained polynomial interpolation with application in trajectory planning, *in* 'Proceedings of the IEEE Symposium on Computer Aided Control System Design, Taipei, Taiwan'. 96
- Herman, A. & Conway, B. [1992], 'Optimal spacecraft attitude control using collocation and nonlinear programming', *Journal of Guidance, Control and Dynamics* **15**(5), 1287–1289. 21
- Herman, A. & Conway, B. [1996], 'Direct optimization using collocation based on high-order gausslobatto quadrature rules', *J. Guidance, Control and Dynamics* **19**(3), 592–599. 20

- Hettich, R. & Kortanek, K. O. [1993], 'Semi-infinite programming : theory, methods, and applications', *SIAM Rev.* **35**(3), 380–429. 58
- Horst, R. & Tuy, H. [1996], *Global Optimization, Deterministic Approaches*, Springer. 57
- Hull, D. G. [1997], 'Conversion of optimal control problems into parameter optimization problems', *J. Guidance, Control and Dynamics* **20**(1). 6, 18
- Hull, D. & Speyer, J. [1982], 'Optimal reentry and plane-change trajectories', *Journal of Astronautical Sciences* **30**(2), 117–130. 17
- Jaulin, L. & Walter, E. [1993], 'Set-inversion via interval analysis for interval analysis for nonlinear bounded-error estimation', *Automatica* **29**, 1053–1064. 146
- Kaplan, W. [1991], *Advanced Calculus*, Addison-Wesley, chapter 5.5, pp. 286–291. 139
- Karlin, S. & Studden, W. [1966], *Tchebychev systems : with applications in analysis and statistics*, Interscience Publishers, John Wiley & Sons. 98
- Kraft, D. [1985], *Computational Mathematical Programming*, Springer, chapter On converting optimal control problems into non linear programming codes, pp. 261–280. 18, 54
- Kvasnica, M., Grieder, P. & Baotić, M. [2004], 'Multi-Parametric Toolbox (MPT)'. 65
- Lastman, G. [1978], 'A shooting method for solving two-point boundary-value problems arising from non-singular bang-bang optimal control problems', *International Journal of control* **27**(4), 513–524. 6
- Lévine, J. [2004a], *Analyse et Commande des Systèmes Non Linéaires*, Ecole des Mines de Paris. 26
- Lévine, J. [2004b], 'On flatness necessary and sufficient conditions', *NOLCOS-Stuttgart* . 26, 31, 32, 33, 34, 38, 118
- Lévine, J. & Ngyuen, D. [2003], 'Flat output characterisation for linear systems using polynomial matrices', *Systems and Control Letters* **48**, 69–75. 51
- Levskii, M. [2002], 'Optimization problem in control over a programmed turn of a spacecraft', *Cosmic Research* **40**, 479–489(11).
- Li, F. & Bainum, P. [1990], 'Numerical approach for solving rigid spacecraft minimum time attitude maneuvers', *Journal of Guidance* **13**(1), 38–45. 7, 14, 16
- Liu, S. & Singh, T. [1996], 'Fuel/time optimal control of spacecraft maneuvers', *Journal of Guidance* **20**(2), 394–397. 23
- Martin, P., Murray, R. & Rouchon, P. [2003], Flat systems, equivalence and trajectography, Technical report, California Institute of Technology CDS. 26
- Meier, E. & Bryson, A. [1990], 'An efficient algorithm for time-optimal control of a two-link manipulator', *Journal of Guidance, Control, and Dynamics* **13**(5), 859–866. 23
- Milam, M. [2003], Real-Time Optimal Trajectory Generation for Constrained Dynamical Systems, PhD thesis, California Institute of Technology. 52
- Milam, M., Mushambi, K. & Murray, R. [2000], A new computational approach to real-time trajectory generation for constrained mechanical systems, in 'IEEE Conference on Decision and Control'.
- Moore, R. [1966], *Interval Arithmetic*, Prentice-Hall. 143, 145

- Morio, V. [2007], Contribution au developpement d'un calcul formel pour la caractérisation des systèmes plats, Technical report, IMS, Université Bordeaux 1. 44
- Neckel, T., Talbot, C. & Petit, N. [2003], 'Collocation and inversion for a reentry optimal control problem', *ICLT-Madrid*. 52
- Nesterov, Y. [2000], *High Performance Optimization.*, Kluwer Academic Press, Dordrecht, chapter Squared functional systems and optimization problems, pp. 405–440. 96, 98, 100
- Neumaier, A. [1990], *Interval Methods for systems of equations*, Cambridge University Press. 146
- Neumaier, A. [1999], 'A simple derivation of the hansen-bliek-rohn-ning-kearfott enclosure for linear interval equation', *Reliable Computing* **5**(2), 131–136. 68
- Neuman, E. [1981], 'Moments and fourier transoms on b-splines', *J. Computational and Applied Mathematics* **7**(1), 51–62. 106
- Ning, S. & Kearfott, R. [1997], 'a comparison of some methods for solving linear interval equations', *SIAM Journal on Numerical Analysis* **34**(4), 1289–1305. 68
- Pesch, H. [1994], 'A pratical guide to the solution of real-life optimal control poblems', *Control and Cyberbetics* **23**, 7–60. 13
- Pontryagin, L., Boltyanskii, V., Gamkrelidze, R. & Mishchenko, E. [1962], *The mathematical Theory of optimal proceses*, Wiley-Interscience. 11
- Quadrat, A. [2007], An introduction to the algebraic theory of linear systems of partial differential equations, Technical report, INRIA. 38
- R., K. [1969], 'Newton-algorithmen zur bestimmung von nullstellen mit fehlerschranken', *Computing* **4**, 187–201. 68
- Renes, J. [1978], On the use of splines and collocation in a trajectory optimization algorithm based on mathematical programming, Technical Report NLR-TR78016U, National Aerospace Lab., The Netherlands. 18, 54
- Roenneke, A., Jansch, C. & Markl, A. [1995], *Advanced Launch and reentry Trajectory Optimization Software (ALTOS) documentation*, European Space Science and Technology Center (ESA), Noordwijk. 7
- Rohn, J. [1993], 'Cheap and tight bounds : therecent result by e. hansen can be made more efficient', *Interval Computation* (4), 13–21. 68
- Rouchon, P. [2001], 'Motion planning, equivalence, infinite dimensional systems', *Int J. Appl. Math. Comput. Sci.* **11**(1), 165–188. 51
- Schaub, H. & Junkins, J. [1996], 'Stéréographic orientation parameters for attitude dynamics : a generalization of the rodrigues parameters', *Journal. Astronautical Sciences* **44**(1), 1–19. 41
- Scrivener, S. & Thomson, R. [1993], 'Time-optimal reorientation of a rigid spacecraft using collocation and nonlinear programming', *Advances in the Astronautical Sciences Proceeding of the AAS/AIAA Astrodynamics Conference* **3**(3), 1905–1924. 7, 21
- Scrivener, S. & Thomson, R. [1994], 'Survey of time-optimal attitude maneuvers', *AIAA Journal of Guidance, Control, and Dynamics* **17**(2), 225–233. 7
- Seywald, H. [1994], 'Trajectory optimization based on differential inclusion', *J. Guidance, Control and Dynamics* **17**(3), 480–487. 19

- Seywald, H. & Kumar, R. [1993], 'Singular control in minimum time spacecraft reorientation', *AIAA Journal of Guidance, Control, and Dynamics* **16**(4), 686–694. 17
- Shen, H. & Tsiotras, P. [1999], 'Time-optimal control of axi-symmetric rigid spacecraft with two controls', *AIAA Journal of Guidance, Control, and Dynamics* **22**(5), 682–694. 7, 23
- Stoer, J. & Bulirsch, R. [1980], *Introduction to numerical analysis*, Springer, New York. 6
- Sussmann, H. & Willems, J. [1997], '300 years of optimal control : from the brachystochrone to the maximum principle', *IEEE Control Syst.* **17**(3), 32–44. 6
- Swartz, B. [1988], 'Conditioning collocation', *SIAM Journal of Numerical Analysis* **25**(1), 124–147. 54
- Toh, K.C. and Todd, M. & Tutuncu, R. [1999], 'Sdpt3 — a matlab software package for semidefinite programming', *Optimization Methods and Software* **11**, 545–581. 110, 112, 114
- Tsiotras, P., Shen, H. & Hall, C. [2001], 'Satellite attitude control and power tracking with momentum wheels', *AIAA Journal of Guidance, Control, and Dynamics* **24**(1), 23–34. 41
- van Nieuwstadt, M. [1997], *Trajectory Generation for Nonlinear Control Systems*, PhD thesis, California Institute of Technology. 52
- van Nieuwstadt, M., Rathinam, M. & Murray, R. [1998], 'Differential flatness and absolute equivalence.', *SIAM J Control and Optimization* **36**. 26
- Villagra, J., d'Andrea Novel, B., Mounier, H. & Pengov, M. [2005], A flatness based control methodology for car-like vehicles : application to obstacle avoiding situations, in 'Journées d'étude Automatique et Automobile', GdR MACS, Bordeaux. 40
- von Stryk, O. & Bulirsch, R. [1992], 'Direct and indirect methods for trajectory optimization', *Annals of Operations Research* **37**, 357–373. 23

---

## Annexe A

# Programmation non linéaire

---

Les techniques de programmation non linéaires sont des techniques itératives de recherche d'optimum. Le principe fondamental de ces techniques, suggéré par Newton, a tout d'abord été utilisé pour trouver les solutions  $x^*$  d'un système algébrique non linéaire  $a(x) = 0$ . L'algorithme de résolution consiste à calculer à partir d'une estimation de la solution  $x$  une nouvelle estimation  $\tilde{x}$  :

$$\tilde{x} = x + \alpha \rho \quad (\text{A.1})$$

où  $\alpha$  est un scalaire et  $\rho$  est la direction de recherche définie par l'équation linéaire :

$$A(x)\rho = -a(x) \quad (\text{A.2})$$

où  $A$  est la matrice définie par :

$$A = \begin{bmatrix} \frac{\partial a_1}{\partial x_1} & \dots & \frac{\partial a_1}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial a_n}{\partial x_1} & \dots & \frac{\partial a_n}{\partial x_n} \end{bmatrix} \quad (\text{A.3})$$

Pour développer la méthodologie générale de l'optimisation non linéaire, posons tout d'abord le problème d'optimisation sans contrainte.

### A.1 Problème d'optimisation non contraint

Supposons que nous devons définir  $n$  variables  $x$  minimisant la fonction de coût scalaire  $J(x)$  :

$$\min_x J(x) \quad (\text{A.4})$$

La condition nécessaire pour que  $x^*$  soit solution de ce problème est :

$$\nabla_{x^*} J = \begin{bmatrix} \frac{\partial J}{\partial x_1^*} \\ \frac{\partial J}{\partial x_2^*} \\ \vdots \\ \frac{\partial J}{\partial x_n^*} \end{bmatrix} = 0 \quad (\text{A.5})$$

Si la méthode utilisée pour trouver  $x^*$  est une recherche linéaire par gradient de descente (méthode de Newton A.2), nous devons calculer une direction de recherche  $p$  telle que :

$$\mathbf{H}(x)\rho = -\nabla_x J \quad (\text{A.6})$$

ou  $\mathbf{H}$  est la matrice hessienne définie par :

$$H(x) = \begin{bmatrix} \frac{\partial^2 J}{\partial x_1^2} & \frac{\partial^2 J}{\partial x_1 \partial x_2} & \cdots & \frac{\partial^2 J}{\partial x_1 \partial x_n} \\ \frac{\partial^2 J}{\partial x_2 \partial x_1} & \frac{\partial^2 J}{\partial x_2^2} & \cdots & \frac{\partial^2 J}{\partial x_2 \partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 J}{\partial x_n \partial x_1} & \frac{\partial^2 J}{\partial x_n \partial x_2} & \cdots & \frac{\partial^2 J}{\partial x_n^2} \end{bmatrix} \quad (\text{A.7})$$

Détaillons maintenant comment les contraintes égalitaires vont être associées au problème que nous venons de présenter.

## A.2 Contraintes égalitaires

Ajoutons au problème précédent  $l$  contraintes égalitaires à satisfaire :

$$\begin{cases} \min_x J(x) \\ c(x) = 0 \end{cases} \quad (\text{A.8})$$

avec  $l \leq n$ . Soit le Lagrangien  $L$  tel que :

$$L(x, \lambda) = J(x) - \lambda^\top c(x) \quad (\text{A.9})$$

Le Lagrangien est une fonction scalaire des  $n$  variables  $x$  et des  $l$  multiplicateurs de Lagrange  $\lambda$ . Les conditions nécessaires d'optimalité pour qu'un point  $(x^*, \lambda^*)$  soit un optimum respectant les contraintes imposent qu'il soit un point stationnaire du Lagrangien défini par :

$$\nabla_x L(x^*, \lambda^*) = \nabla_x J(x^*) - C^\top(x^*)\lambda = 0 \quad (\text{A.10})$$

$$\text{et } \nabla_\lambda L(x^*, \lambda^*) = -c(x^*) = 0 \quad (\text{A.11})$$

avec

$$C(x) = \begin{bmatrix} \frac{\partial c_1}{\partial x_1} & \frac{\partial c_1}{\partial x_2} & \cdots & \frac{\partial c_1}{\partial x_n} \\ \frac{\partial c_2}{\partial x_1} & \frac{\partial c_2}{\partial x_2} & \cdots & \frac{\partial c_2}{\partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial c_l}{\partial x_1} & \frac{\partial c_l}{\partial x_2} & \cdots & \frac{\partial c_l}{\partial x_n} \end{bmatrix} \quad (\text{A.12})$$

Les  $(n + m)$  variables  $(x, \lambda)$  qui satisfont les conditions d'optimalité (A.10) et (A.11) vont être obtenues par une méthode analogue à celle de Newton (A.1) dont la direction de recherche  $p$  est définie par :

$$\begin{bmatrix} \mathbf{H}_L & C^\top \\ C & 0 \end{bmatrix} \begin{bmatrix} \rho \\ -\lambda \end{bmatrix} = \begin{bmatrix} -\nabla_x F \\ -c \end{bmatrix} \quad (\text{A.13})$$

où  $\mathbf{H}_L$  représente le Hessien du Lagrangien tel que

$$\mathbf{H}_L = \nabla_x^2 F - \sum_{i=1}^m \lambda_i \nabla_x^2 c_i \quad (\text{A.14})$$

Le système linéaire (A.13) est connu comme les équations de Kuhn-Tucker ou encore de Karush-Kuhn-Tucker. On peut également noter que la direction de descente  $p$  définie par la résolution du problème (A.13) est aussi solution du problème de minimisation quadratique sous contraintes :

$$\begin{aligned} \min_{\rho} \frac{1}{2} \rho^\top \mathbf{H}_L \rho + \nabla_x J^\top \rho \\ \text{sous } C\rho = -c \end{aligned} \quad (\text{A.15})$$

Ce dernier problème nommé sous-problème d'optimisation quadratique est très important dans la résolution des problèmes d'optimisation sous contraintes inégales.

### A.3 Contraintes inégales

Considérons maintenant le problème d'optimisation sous  $l$  contraintes inégales :

$$\begin{cases} \min_x J(x) \\ c(x) \geq 0 \end{cases} \quad (\text{A.16})$$

Notons que le nombre de contraintes  $l$  est dorénavant libre contrairement aux hypothèses du paragraphe précédent.

A l'instar des contraintes inégales prises en compte dans la résolution du problème de commande optimale par les conditions nécessaires d'optimalité (cf le paragraphe I.3.1.3), l'ensemble des contraintes va être classé en deux sous-ensembles :

- les contraintes inactives i.e. les contraintes strictement satisfaites,  $c_i(x) > 0$ ,
- les contraintes actives i.e. les contraintes qui sont sur leurs frontières ou non respectées,  $c_j(x) \leq 0$

Les contraintes inactives sont ignorées et les contraintes actives sont remplacées par les contraintes égales  $c_j(x) = 0$ . Toute la difficulté réside dans la détermination de ces deux sous-ensembles. En effet, s'ils sont parfaitement connus, alors la méthode de résolution consiste à résoudre le problème d'optimisation sous contraintes égales défini plus haut en ne considérant que les contraintes actives. Dans le cas contraire, les algorithmes de résolution deviennent plus complexes puisqu'ils nécessitent de résoudre une succession de résolutions des équations de Kuhn-Tucker (A.13) via le sous-problème d'optimisation quadratique (A.15) avec une discrimination des contraintes actives et inactives à chaque itération. Ainsi, la plupart des moteurs d'optimisations disponibles sont basés sur le principe de résolutions successives du sous-problème d'optimisation quadratique sous contraintes linéaires (A.15). On dénomme ces algorithmes SQP pour *sequential quadratic programming*.

La description des algorithmes d'optimisation présentée ici ne se prétend pas être exhaustive. En effet, pour des raisons de clarté, nous n'avons pas introduit les conditions suffisantes d'optimalité obtenues à partir des conditions des Karush-Kuhn-Tucker de second ordre. Une discussion complète sur l'apport et l'utilisation de ces conditions suffisantes est disponible dans [Gil, Murray & Wright 1981].



---

## Annexe B

# Eléments de géométrie différentielle

---

### B.1 Morphismes

Soit  $X$  et  $Y$  deux espaces.

#### Définition B.1

Une application  $f : X \rightarrow Y, x \mapsto y$  est continue au point  $x$  si, pour tout voisinage  $V_y$  de  $f(x)$ ,  $f^{-1}(V_y)$  est un voisinage de  $x$ . Elle est continue sur  $X$ , si elle est continue en tout point de  $X$ .

#### Définition B.2 (Homéomorphisme)

Un homéomorphisme  $f$  de  $X$  vers  $Y$  est une bijection telle que  $f$  et  $f^{-1}$  sont continues

Soit  $E$  et  $F$  des espaces vectoriels normés.

#### Définition B.3

Une application  $f : E \rightarrow F$  est un isomorphisme si :

1.  $f$  est linéaire continue
2. il existe une application linéaire continue  $g : F \rightarrow E$  telle que :

$$g \circ f = I_E \text{ et } f \circ g = I_F \tag{B.1}$$

où  $I$  est l'identité.

Soit  $U$  et  $V$  deux ouverts de  $E$  et  $F$  respectivement.

#### Définition B.4

Une application  $f$  de  $U$  dans  $V$  est un  $C^q$ -difféomorphisme si  $f$  est une bijection  $U \rightarrow V$  est de classe  $C^q$  et si  $f^{-1} : V \rightarrow U$  est de classe  $C^q$ .

## B.2 Théorèmes des fonctions implicites

**Théorème B.1 (Théorème des fonctions implicites)** Soit  $\varphi$  une application  $k$  fois continûment différentiable, avec  $k \geq 1$ , d'un ouvert  $U \subset \mathbb{R}^n$  dans  $\mathbb{R}^{n-p}$  avec  $0 \leq p \leq n$ , et supposons qu'il existe au moins un  $x_0 \in U$  tel que  $\varphi(x_0) = 0$ . Si en chaque point de  $x$  de  $U$  l'application linéaire tangente  $D\varphi(x)$  est de rang plein égal à  $n-p$ , il existe un voisinage  $V = V_1 \times V_2 \subset U$  de  $\mathbb{R}^n = \mathbb{R}^p \times \mathbb{R}^{n-p}$  et une application  $k$  fois continûment différentiable  $\psi$  de  $V_1$  dans  $V_2$  tels que les deux ensembles  $\{x \in V_1 \times V_2 \mid \varphi(x) = 0\}$  et  $\{(x_1, x_2) \in V_1 \times V_2 \mid x_2 = \psi(x_1)\}$  sont égaux. ■

Autrement dit, la fonction  $\psi$  vérifie localement  $\varphi(x_1, \psi(x_1)) = 0$  et la variable  $x_2 = \psi(x_1)$  est décrite à l'aide des  $p$  variables indépendantes de  $x_1$ .

## B.3 Variété différentielle

Soit  $V$  un ensemble d'éléments appelés points et  $F$  un espace vectoriel, normé réel et de dimension finie  $n$ .  $V$  est munie d'une topologie i.e. tout point de  $V$  appartient à au moins un ouvert  $U_i$  de  $V$  (notion de recouvrement).

### Définition B.5 (Carte)

Une carte locale sur  $V$  est le couple noté  $(U_i, \phi_i)$ , formé d'un ouvert de  $V$  et d'un homéomorphisme  $\phi$  de  $U_i$  vers un sous-ensemble ouvert  $\phi(U_i)$  de  $F$ .

### Définition B.6 (Coordonnées locales)

Les coordonnées locales d'un point  $p$  du domaine  $U$  d'une carte  $(U, \phi)$  de  $V$  sont les coordonnées du point  $\phi(p)$  de  $\mathbb{R}^n$ . On note

$$(x_1, \dots, x_n)$$

le  $n$ -uplet de réels attachés au point  $p$ . La bijection  $\phi$  associe à tout point  $p \in U \subset V$  le  $n$ -uplet de réels  $(x_1, \dots, x_n)$ . Réciproquement,  $\phi^{-1}$  associe à tout  $n$ -uplet de réels un point de  $U$ . Une carte locale définit un système de coordonnées locales.

### Définition B.7

Deux cartes  $(U_i, \phi_i)$  et  $(U_j, \phi_j)$  sur  $V$ , tel que  $U_i \cap U_j \neq \emptyset$  sont  $C^q$ -compatibles ( $q \geq 1$ ) si

$$\phi_{ji} = \phi_j \circ \phi_i^{-1} \quad (\text{B.2})$$

est un difféomorphisme de classe  $C^q$  entre les ouverts  $\phi_i(U_i \cap U_j)$  et  $\phi_j(U_i \cap U_j)$  de  $\mathbb{R}^n$

### Définition B.8 (Atlas)

Un atlas de classe  $C^q$  sur  $V$  est un ensemble  $A$  de cartes  $(U_i, \phi_i)$  telles que :

- les domaines  $U_i$  de cartes constituent un recouvrement de  $V$ ,
- les cartes  $(U_i, \phi_i)$  et  $(U_j, \phi_j)$  telles que  $U_i \cap U_j \neq \emptyset$  de  $A$  sont  $C^q$ -compatibles.

### Définition B.9 (Atlas maximal et structure de variété différentiable)

L'atlas maximal  $\bar{A}$  est l'atlas se composant de toutes les cartes compatibles avec  $A$ . Un atlas maximal sur  $V$  munit  $V$  d'une structure de variété différentiable.

**Définition B.10 (Variété différentiable)**

Une variété différentiable de classe  $C^q$  est un couple formé d'un espace topologique et d'un atlas maximal :

$$(V, \bar{A})$$

Sa dimension étant  $n$ , on note la variété  $(V_n, \bar{A})$  plus simplement  $V_n$ .

**Définition B.11 (Variété produit)**

La variété produit  $X_n \times Y_m$  de deux variétés est la variété définie par l'atlas produit des atlas de  $X_n$  et de  $Y_m$ . Sa dimension est la somme des dimensions de chacune des variétés  $X_n$  et  $Y_m$ .

Soit  $p$  un point d'une variété différentiable  $V_n$ .

**Définition B.12 (Espace vectoriel tangent)**

L'espace vectoriel tangent est l'ensemble des vecteurs tangents à  $V_n$  au point  $p$ . Il est noté  $T_p V_n$ . Un vecteur tangent au point  $p$  a pour expression :

$$X_p = \left[ X_1 \frac{\partial}{\partial x_1} \quad \dots \quad X_n \frac{\partial}{\partial x_n} \right]^T \quad (\text{B.3})$$

La base de cet espace tangent est formée par les  $n$  opérateurs  $\frac{\partial}{\partial x_i}$  qui se définissent par :

$$\frac{\partial}{\partial x_i} : U \rightarrow \mathbb{R} : g \mapsto \frac{\partial g}{\partial x_i} \quad (\text{B.4})$$

L'espace vectoriel tangent est donc de dimension  $n$ .

## B.4 Fibré tangent, Champ de vecteurs et Dérivée de Lie

Considérons maintenant la réunion de tous les espaces tangents à  $V_n$  en tout point de  $V_n$ .

**Définition B.13**

Le fibré tangent à  $V_n$  est la variété :

$$TV_n = \bigcup_{x \in V_n} (x \times T_x V_n) \quad (\text{B.5})$$

Le fibré tangent  $TV_n$  possède une structure naturelle de variété différentiable de dimension  $2n$ . La base du fibré tangent est  $V_n$ .

**Définition B.14 (Champ de vecteurs)**

Un champ de vecteurs sur  $V_n$  est une application  $X$  telle que :

$$X : V_n \rightarrow TV_n : x \mapsto X(x) = (x, X_x) \quad (\text{B.6})$$

Cette application associe à tout point de  $x$  de  $V_n$  un couple formé du point et d'un vecteur tangent  $X_x$  en ce point. Un champ de vecteurs est différentiable si l'application qui le définit est de classe  $C^\infty$ . L'ensemble des champs de vecteurs différentiables sur  $V_n$  est noté  $\mathcal{X}(V_n)$ .

Soit  $V, W \in \mathcal{X}(V_n)$ .

**Définition B.15**

Le produit des champs de vecteurs  $V, W$  est un opérateur défini par :

$$\forall g \in C^\infty : (VW)g = V(Wg) \quad (\text{B.7})$$

**Définition B.16 (Crochet de Lie)**

L'opération crochet de Lie est l'application :

$$[\ ] : \mathcal{X}(V_n) \times \mathcal{X}(V_n) \rightarrow \mathcal{X}(V_n) : (V, W) \mapsto [V, W] = VW - WV \quad (\text{B.8})$$

Le crochet des champs de vecteurs  $V$  et  $W$  opère sur  $C^\infty(V_n)$ , l'ensemble des fonctions infiniment différentiables sur  $V_n$  :

$$[V, W] : C^\infty(V_n) \rightarrow C^\infty(V_n) : g \mapsto [V, W]g = V(Wg) - W(Vg) \quad (\text{B.9})$$

Le crochet de Lie permet d'introduire la notion de dérivée de Lie.

**Définition B.17 (Dérivée de Lie)**

La dérivée de Lie du champ de vecteurs  $V$  par rapport au champ de vecteurs  $W$  est définie par :

$$L_W V = [W, V] \quad (\text{B.10})$$

En introduisant les coordonnées locales  $x$ , il vient :

$$L_W V = \sum_{i=1}^n W_i(x) \frac{\partial V}{\partial x_i}(x) \quad (\text{B.11})$$

**B.5 Fibré cotangent et champ de 1-formes****Définition B.18 (Espace vectoriel cotangent)**

L'espace vectoriel cotangent, en  $x$ , à  $V_n$  est le dual de l'espace tangent  $T_x V_n$  et il est noté  $T_x^* V_n$ . Il a pour repère la base formée par les  $n$  opérateur  $dx_i$  qui sont la différentielle de la projection  $x \mapsto x_i$  et qui se définit par :

$$\langle dx_i, \frac{\partial}{\partial x_j} \rangle = \delta_{i,j} \quad (\text{B.12})$$

$\delta_{i,j}$  étant le symbole de Kroneker.

**Définition B.19 (1-forme)**

Une 1-forme au point  $x$  de  $V_n$  est une forme linéaire sur  $T_x V_n$  que l'on note généralement  $\omega_x$ . La dualité des 1-formes et vecteurs tangents est mise en évidence par l'égalité :

$$\langle \omega_x, V \rangle = \omega_x(V) = V(\omega_x) \quad (\text{B.13})$$

Une 1-forme  $\omega_x$  s'écrit dans le repère de  $T_x^* V_n$  :

$$\omega_x = \sum_{i=1}^n \omega_i dx_i \quad (\text{B.14})$$

**Définition B.20**

Une 1-forme  $\omega$  sur  $V_n$  est fermée si sa différentielle extérieure est nulle :  $d\omega = 0$

---

## Annexe C

# Espaces de fonctions polynomiales par morceaux et B-splines

---

### C.1 Espaces de fonctions polynomiales par morceaux et B-splines

Dans cette annexe, nous allons définir les B-splines comme une base de représentation de l'espace des polynômes par morceaux. Les définitions et propriétés présentées ici sont classiques. Cependant, le premier objectif de cette annexe est de poser le vocabulaire et les notations qui seront utilisés dans l'ensemble du mémoire. Le second est de rappeler que les B-splines constituent une base de représentation des fonctions polynomiales par morceaux.

#### Définition C.1 (Fonction polynomiale par morceaux)

Soient  $\Delta$  un intervalle de  $\mathbb{R}$ , une séquence de points strictement croissante  $\xi = \{\xi_i\}_{i=1}^{l+1}$  appelée séquence de segmentation,  $k$  un entier naturel positif et  $\{P_1, \dots, P_l\}$  une séquence de  $l$  polynômes d'ordre  $k$  (i.e. de degré inférieur à  $k$ ). On définit  $f$ , une fonction polynomiale par morceaux ou polynôme par morceaux d'ordre  $k$  sur la séquence  $\xi$  telle que :

$$f(x) = \begin{cases} P_1(x) & \text{si } \xi_1 < x < \xi_2 \\ \vdots & \\ P_l(x) & \text{si } \xi_l < x < \xi_{l+1} \\ 0 & \text{si } x < \xi_1 \text{ ou } \xi_{l+1} < x \end{cases} \quad (\text{C.1})$$

On appelle les points  $\xi$  les noeuds de segmentation de  $f$ .

Notons que  $f$ , aux noeuds intérieurs de segmentation  $\{\xi_i\}_{i=2}^l$ , n'est définie que si les valeurs de  $f$  à gauche et à droite du point sont égales.

On dit que deux fonctions polynomiales par morceaux peuvent être coordonnées si elles possèdent la même séquence de segmentation  $\xi$ .

**Définition C.2 (Espace des polynômes par morceaux coordonnés d'ordre  $k$ )**

On note  $\mathbb{P}_{k,\xi}$  l'espace linéaire des polynômes par morceaux d'ordre  $k$  comme étant la collection de tous les polynômes par morceaux d'ordre  $k$  définis sur la séquence de segmentation  $\xi$ . Cet espace est de dimension  $kl$  au sens où il faut déterminer les  $k$  coefficients des  $l$  polynômes pour obtenir un polynôme par morceaux.

Donnons maintenant la définition du sous-espace de  $\mathbb{P}_{k,\xi}$  des fonctions polynomiales par morceaux respectant des contraintes de continuité aux points de segmentation.

**Définition C.3 (Espace des polynômes par morceaux d'ordre  $k$  et de continuité  $v_i$  en  $\xi_i$ )**

Soit  $\mathbb{P}_{k,\xi,v}$ , un sous-espace de  $\mathbb{P}_{k,\xi}$ , défini comme l'ensemble des fonctions polynomiales par morceaux vérifiant les conditions de classe de continuité  $C^{v_i}$  en  $\xi_i$ . Soit  $v$  la séquence des conditions de continuité  $(v_1, \dots, v_{l+1})$ , avec  $v_i < k$ ,  $i = 1, \dots, l+1$ . Ces contraintes s'écrivent :

$$\lim_{\substack{x \rightarrow \xi_i \\ x < \xi_i}} f^{(j-1)}(x) = \lim_{\substack{x \rightarrow \xi_i \\ x > \xi_i}} f^{(j-1)}(x), \quad \text{avec } j = 1, \dots, v_i \text{ et } i = 2, \dots, l \quad (\text{C.2})$$

**Remarque C.1** Etant un sous-espace de l'espace linéaire  $\mathbb{P}_{k,\xi}$ , l'espace  $\mathbb{P}_{k,\xi,v}$  est aussi un espace linéaire.

Définissons maintenant les B-splines d'ordre  $k$  comme une famille de polynômes par morceaux de  $\mathbb{P}_{k,\xi,v}$ .

**Définition C.4 (B-splines d'ordre  $k$ )**

Soit  $T$  une séquence non décroissante de points, appelée séquence nodale, basée sur la séquence  $\xi = \{\xi_i\}_{i=1}^{l+1}$  fractionnant  $\Delta$  en  $l$  segments telle que :

$$\begin{aligned} T &= \{t_1, \dots, t_p\} \\ &= \underbrace{\{\xi_1, \dots, \xi_1\}}_{\rho_1 \text{ fois}} \underbrace{\{\xi_2, \dots, \xi_2\}}_{\rho_2 \text{ fois}} \dots \underbrace{\{\xi_{l+1}, \dots, \xi_{l+1}\}}_{\rho_{l+1} \text{ fois}} \end{aligned} \quad (\text{C.3})$$

La famille  $S$  des B-splines d'ordre  $k$  construite sur la séquence nodale  $T$  est déterminée par l'algorithme itératif de Cox-de Boor [de Boor 1978] :

$$\begin{cases} B_{j,0}(x) = \begin{cases} 1 & t_j \leq x \leq t_{j+1} \\ 0 & \text{ailleurs} \end{cases} \\ B_{j,k}(x) = \frac{x-t_j}{t_{j+k-1}-t_j} B_{j,k-1}(x) + \frac{t_{j+k}-x}{t_{j+k}-t_{j+1}} B_{j+1,k-1}(x) \end{cases} \quad (\text{C.4})$$

Et  $\zeta(B_{i,k})$  désigne le support du  $i^{\text{me}}$  élément de  $S$  sur lequel l'élément  $B_{i,k}$  est non nul :

$$\zeta(B_{i,k}) = [\underline{\zeta}(B_{i,k}), \bar{\zeta}(B_{i,k})] = [t_j, t_{j+k}] \quad (\text{C.5})$$

**Remarque C.2** Notons que, par construction, chaque B-spline  $B_{i,k}$  est positive ou nulle sur son support.

Rappelons maintenant le théorème qui établit la famille de B-splines d'ordre  $k$  construite sur la séquence  $T$  comme une base du sous-espace linéaire  $\mathbb{P}_{k,\xi,v}$ .

**Théorème C.1 (Théorème de Curry-Schoenberg)** Soient la séquence  $\xi$  définie comme précédemment, et la séquence des conditions de continuité  $v = \{v_i\}_{i=1}^{l+1}$  vérifiant  $\forall i, v_i \leq k$  telles que

$$m = k + \sum_{i=2}^l (k - v_i) = kl - \sum_{i=2}^l v_i = \dim(\mathbb{P}_{k,\xi,v}) \quad (\text{C.6})$$

et soit la séquence nodale  $T$  définie comme précédemment avec

$$\rho_i = \begin{cases} k & \text{si } i = 1, l+1 \\ k - v_i & \text{si } i = 2, \dots, l \end{cases} \quad (\text{C.7})$$

Il vient alors que la famille de B-splines  $\{B_{1,k}, \dots, B_{m,k}\}$  d'ordre  $k$  construite sur la séquence  $T$  est une base de  $\mathbb{P}_{k,\xi,v}$ . ■

La démonstration de ce théorème a été développée dans [de Boor 1978, pages 115-116]. Nous en donnons ici les idées directrices.

### Démonstration

La démonstration de ce théorème se base sur deux éléments :

1. le nombre d'éléments de la famille  $\mathcal{S}$  est égal à la dimension de  $\mathbb{P}_{k,\xi,v}$
2. la famille  $\mathcal{S}$  constitue une famille libre de fonctions.

Dans un premier temps, il nous faut prouver que chaque B-spline  $B_{i,k}$  appartient à  $\mathbb{P}_{k,\xi,v}$  sur l'intervalle  $\Delta$ . Tout d'abord, il est admis qu'une B-spline  $B_{i,k}$  est un polynôme par morceaux d'ordre  $k$  défini sur la séquence nodale  $\{t_i, \dots, t_{i+k}\}$ . Ensuite, la dérivée  $s^{\text{ème}}$  de la B-spline  $B_{i,k}$  est discontinue au point  $\xi_j$ , s'il existe un indice  $r \in [i, i+k]$  tel que  $t_r = \xi_j$  et  $k - 1 - j_r = s$ .  $j_r$  correspond au nombre d'éléments de la séquence  $t_i, \dots, t_{i+k}$  tel que  $t_m = t_r$  avec  $i \leq m \leq r$ . Il s'ensuit que  $j_r$  est inférieur à  $k - v_j$ , qui est le nombre d'éléments de la séquence  $T$  égaux à  $\xi_j$ . On en déduit que  $s \geq v_j$  et que les  $v_j - 1$  premières dérivées de la B-spline  $B_{i,k}$  sont continues en  $t_r$ . Nous venons de prouver l'appartenance des  $m$  B-splines  $B_{i,k}$  à  $\mathbb{P}_{k,\xi,v}$ . Ainsi, le nombre d'éléments de la famille  $\mathcal{S}$  est égal à la dimension de  $\mathbb{P}_{k,\xi,v}$ .

Il est maintenant nécessaire de démontrer que la famille de B-splines  $\mathcal{S}$  est une famille libre pour finir la démonstration de ce théorème. La démonstration de ce point étant relativement fastidieuse et technique, nous renvoyons le lecteur intéressé à la référence [de Boor & Fix 1973]. □



---

## Annexe D

# Superellipsoïdes : définitions complémentaires

---

### D.1 Volume d'une superellipsoïde

Nous nous proposons de définir le volume d'une superellipsoïde. Pour ce faire, nous allons dans un premier temps déterminer l'aire d'une superellipse, puis en déduire le volume de la superellipsoïde par calcul infinitésimal.

Soit une superellipse  $E_1$  donnée par l'équation (III.25). L'aire  $A$  de  $E_1$  est définie par la formule de Green [Kaplan 1991] :

$$A(E_1) = \frac{1}{2} \oint_{C'} (x_1 dx_2 - x_2 dx_1) \quad (D.1)$$

où  $C'$ , le chemin d'intégration, est le contour de la superellipse. Le calcul aboutit à l'intégrale suivante :

$$\begin{aligned} A(E_1) &= 2a_1 a_2 \varepsilon_1 \int_0^{\frac{\pi}{2}} (\sin^{\varepsilon_1-1} \theta_1 \cos^{\varepsilon_1+1} \theta_1 + \sin^{\varepsilon_1+1} \theta_1 \cos^{\varepsilon_1-1} \theta_1) d\theta_1 \\ &= a_1 a_2 \varepsilon_1 \left[ B\left(\frac{\varepsilon_1}{2}, \frac{\varepsilon_1+2}{2}\right) + B\left(\frac{\varepsilon_1}{2}, \frac{\varepsilon_1+2}{2}\right) \right] \\ &= 2a_1 a_2 \varepsilon_1 B\left(\frac{\varepsilon_1}{2}, \frac{\varepsilon_1+2}{2}\right) \end{aligned} \quad (D.2)$$

La fonction  $B(x, y)$  est définie par :

$$B(x, y) = 2 \int_0^{\frac{\pi}{2}} \sin^{2x-1} \phi \cos^{2y-1} \phi d\phi = \frac{\Gamma(x)\Gamma(y)}{\Gamma(x+y)} \quad (D.3)$$

où la fonction classique  $\Gamma$  est définie selon la formule d'Euler :

$$\Gamma(x) = \lim_{n \rightarrow \infty} \frac{n! n^x}{(x+1) \dots (z+n)} \quad (D.4)$$

Afin d'obtenir le volume de  $\mathcal{S}_3$ , une superellipsoïde de dimension 3, nous allons d'abord écrire l'expression d'un élément infinitésimal de volume  $dV$ . L'aire de la tranche de la superellipsoïde à la coordonnée  $z$  s'écrit en fonction des paramètres de la seconde superellipse  $E_2$  : l'élément infinitésimal de volume  $dV$  s'écrit à partir de l'aire de la tranche de la superellipsoïde à la coordonnée  $x_3$  :

$$\begin{aligned} dV &= A(x_3)dx_3 \\ &= A(\theta_2) \frac{\partial x_3}{\partial \theta_2} d\theta_2 \end{aligned} \quad (\text{D.5})$$

avec

$$\frac{\partial x_3}{\partial \theta_2} = a_3 \varepsilon_2 \sin^{\varepsilon_2-1} \theta_2 \cos \theta_2 \quad (\text{D.6})$$

$$A(\theta_2) = 2a(\theta_2)b(\theta_2)\varepsilon_1 B\left(\frac{\varepsilon_1}{2}, \frac{\varepsilon_1+2}{2}\right) \quad (\text{D.7})$$

où

$$a(\theta_2) = a_1 \cos^{\varepsilon_2} \theta_2 \quad (\text{D.8})$$

$$b(\theta_2) = a_2 \cos^{\varepsilon_2} \theta_2 \quad (\text{D.9})$$

On obtient l'expression de  $dV$  :

$$dV = 2a_1 a_2 a_3 \varepsilon_1 \varepsilon_2 B\left(\frac{\varepsilon_1}{2}, \frac{\varepsilon_1+2}{2}\right) \cos^{2\varepsilon_2+1} \theta_2 \sin^{\varepsilon_2-1} \theta_2 d\theta_2 \quad (\text{D.10})$$

Il vient ainsi le volume de  $\mathcal{S}_3$  :

$$\begin{aligned} V_3 &= \int dV \\ &= \int_0^{\frac{\pi}{2}} 2a_1 a_2 a_3 \varepsilon_1 \varepsilon_2 B\left(\frac{\varepsilon_1}{2}, \frac{\varepsilon_1+2}{2}\right) \cos^{2\varepsilon_2+1} \theta_2 \sin^{\varepsilon_2-1} \theta_2 d\theta_2 \\ &= 2a_1 a_2 a_3 \varepsilon_1 \varepsilon_2 B\left(\frac{\varepsilon_1}{2}, \frac{\varepsilon_1+2}{2}\right) B\left(\frac{\varepsilon_2}{2}, \frac{2\varepsilon_2+2}{2}\right) \end{aligned} \quad (\text{D.11})$$

Le résultat précédent se généralise au cas d'une superellipsoïde de dimension  $n$  :

$$V_n = 2a_n \prod_{i=1}^{n-1} a_i \varepsilon_i B\left(\frac{\varepsilon_i}{2}, \frac{i}{2} \varepsilon_1 + 1\right) \quad (\text{D.12})$$

**D.1.0.0.1 Fonction d'appartenance** La fonction d'appartenance  $\mathcal{F}_n$  à une superellipsoïde  $\mathcal{S}_n$  se définit à partir de l'équation implicite de sa surface. En deux dimensions, les coordonnées d'une superellipse sont données par

$$E_1(\varepsilon_i, \theta_i) = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} a_1 \cos^{\varepsilon_1} \theta_1 \\ a_2 \sin^{\varepsilon_1} \theta_1 \end{bmatrix}, \quad \theta_1 = [-\pi, \pi] \quad (\text{D.13})$$

On peut écrire :

$$(\cos^{\varepsilon_1} \theta_1)^{\frac{2}{\varepsilon_1}} + (\sin^{\varepsilon_1} \theta_1)^{\frac{2}{\varepsilon_1}} = \cos^2 \theta_1 + \sin^2 \theta_1 \quad (\text{D.14})$$

il vient ainsi

$$\frac{x_1^{\frac{2}{\varepsilon_1}}}{a_1} + \frac{x_2^{\frac{2}{\varepsilon_1}}}{a_2} = 1 \quad (\text{D.15})$$

On peut écrire la fonction d'appartenance  $\mathcal{F}_2$  telle que :

$$\mathcal{F}_2(x_1, x_2) = x_1^{\frac{2}{\varepsilon_1}} + x_2^{\frac{2}{\varepsilon_1}} \quad (\text{D.16})$$

On peut alors établir les propriétés suivantes :

- si  $\mathcal{F}_2(x_1, x_2) = 1$  alors le point  $(x_1, x_2)$  est un point de la courbe
- si  $\mathcal{F}_2(x_1, x_2) < 1$  alors le point  $(x_1, x_2)$  est à l'intérieur de la superellipse
- si  $\mathcal{F}_2(x_1, x_2) > 1$  alors le point  $(x_1, x_2)$  est à l'extérieur de la superellipse

Ces trois règles restent valables quelque soit la dimension considérée. Il nous faut maintenant déterminer les fonctions d'appartenance aux superellipsoïdes en dimension  $n$ .

Considérons un point  $P$  en dimension  $n$  de coordonnées  $[x_1, \dots, x_n]$ . La norme  $\Lambda_{n,n}$  associée à la superellipsoïde est obtenue par la relation de récurrence suivante

$$\begin{cases} \Lambda_{n,2}(x) = \left(\frac{x_1}{a_1}\right)^{\frac{2}{\varepsilon_1}} + \left(\frac{x_2}{a_2}\right)^{\frac{2}{\varepsilon_1}} \\ \Lambda_{n,k}(x) = (\Lambda_{n,k-1})^{\frac{\varepsilon_{k-2}}{\varepsilon_{k-1}}} + \left(\frac{x_k}{a_k}\right)^{\frac{2}{\varepsilon_{k-1}}} \end{cases} \quad (\text{D.17})$$

Si  $P$  est un point de la surface de la superellipsoïde, on a alors :

$$\Lambda_{n,n}(x) = 1 \quad (\text{D.18})$$

En effet, on a :

$$\Lambda_{n,2}(x) = (\cos^2 \theta_1 + \sin^2 \theta_1) \prod_{i=2}^{n-1} \cos^{\frac{2\varepsilon_i}{\varepsilon_1}} \theta_i$$

La récurrence précédente nous donne :

$$\Lambda_{n,k}(x) = (\cos^2 \theta_{k-1} + \sin^2 \theta_{k-1}) \prod_{i=k}^{n-1} \cos^{\frac{2\varepsilon_i}{\varepsilon_{k-1}}} \theta_i$$

Il vient que pour  $k = n$  nous avons :

$$\begin{aligned} \Lambda_{n,n} &= (\Lambda_{n-1})^{\frac{\varepsilon_{n-2}}{\varepsilon_{n-1}}} + \left(\frac{x_n}{a_n}\right)^{\frac{2}{\varepsilon_{n-1}}} \\ &= \cos^2 \theta_{n-1} + \sin^2 \theta_{n-1} \\ &= 1 \end{aligned}$$

Ainsi, nous pouvons établir la fonction d'appartenance  $\mathcal{F}_n$  définie par :

$$\mathcal{F}_n(x, a) = \Lambda_{n,n}, \quad x, a \in \mathbb{R}^n \quad (\text{D.19})$$



---

## Annexe E

# Analyse par intervalles et inversion ensembliste

---

### E.1 Analyse par intervalles

Nous proposons ici une introduction à l'analyse par intervalles. On doit à Moore les premiers développements de l'analyse par intervalles [Moore 1966]

#### Définition E.1 (Intervalles sur $\mathbb{R}$ )

Un intervalle, noté par  $x$ , est un ensemble connexe et borné de  $\mathbb{R}$  qui se définit par :

$$x = [x^-, x^+] = \{x \in \mathbb{R} | x^- \leq x \leq x^+\} \quad (\text{E.1})$$

Les bornes réelles  $x^-$  et  $x^+$  sont respectivement les bornes inférieures et supérieures de  $x$ . Un intervalle est dit dégénéré lorsque  $x^- = x^+$ . L'ensemble des intervalles de  $\mathbb{R}$  est noté par  $\mathbb{IR}$ .

Les opérations mathématiques élémentaires sont étendues aux intervalles. Le résultat d'une opération entre deux intervalles est un intervalle qui contient tous les résultats des opérations entre les éléments des deux intervalles. Le résultat d'une opération entre deux intervalles de bornes finies est obtenu en travaillant uniquement sur leurs bornes.

#### Définition E.2 (Opérations élémentaires)

Soient  $x, y \in \mathbb{IR}$ , et  $\circ \in \{+, -, *, /\}$ , alors :

$$x \circ y = \{x \circ y | x \in x, y \in y\} \quad (\text{E.2})$$

La définition E.2 est valable pour toutes les opérations à l'exception de la division lorsque  $0 \in y$ . Dans ce dernier cas, le résultat n'est pas un intervalle. Détaillons l'équation (E.2) pour les quatre

opérations élémentaires :

$$x + y = x^- + y^-, x^+ + y^+ \quad (\text{E.3})$$

$$x - y = x^- - y^+, x^+ - y^- \quad (\text{E.4})$$

$$x * y = [\min(x^- y^-, x^- y^+, x^+ y^-, x^+ y^+), \dots, \max(x^- y^-, x^- y^+, x^+ y^-, x^+ y^+)] \quad (\text{E.5})$$

$$x/y = x * 1/y^+, 1/y^-, \text{ si } 0 \notin y \quad (\text{E.6})$$

Nous pouvons aussi définir d'autres grandeurs caractéristiques d'un intervalle  $x \in \mathbb{R}$  :

- sa largeur :  $w(x) = x^+ - x^- \geq 0$
- son milieu  $mid(x) = (x^+ + x^-)/2$
- son rayon  $rad(x) = (x^+ - x^-)/2 \geq 0$

### Définition E.3

Un vecteur d'intervalles (ou pavé), noté par  $\mathbf{x}$ , se définit par :

$$\mathbf{x} = \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix} \quad (\text{E.7})$$

Les éléments du vecteur  $\mathbf{x}$  appartiennent à  $\mathbb{IR}$ . On note par  $\mathbb{IR}^n$  l'ensemble des vecteurs d'intervalles de  $\mathbb{R}^n$ .

On définit les fonctions élémentaires suivantes : soit  $\mathbf{x} \in \mathbb{IR}^n$ , alors

- la borne inférieure est :  $\inf(\mathbf{x}) = (x_1^-, \dots, x_n^-)^T$
- la borne supérieure est :  $\sup(\mathbf{x}) = (x_1^+, \dots, x_n^+)^T$
- la largeur est :  $w(\mathbf{x}) = \max_{j=1}^n (x_j^+ - x_j^-)$
- le milieu est :  $mid(\mathbf{x}) = (mid(x_1), \dots, mid(x_n))^T$

Définissons maintenant l'image d'un vecteur d'intervalle par une application.

### Définition E.4 (fonction d'inclusion)

Soit  $f : \mathbb{R}^n \rightarrow \mathbb{R}^m$  une fonction vectorielle contenant un nombre fini d'opérations arithmétiques et de fonctions élémentaires  $\{\cos, \sin, \log, \dots\}$ . Une fonction d'inclusion de  $f$ , notée par  $\mathbf{f}$ , est une fonction de  $\mathbb{IR}^n$  dans  $\mathbb{IR}^m$  vérifiant

$$f(\mathbf{x}) = \{f(x) | x \in \mathbf{x}\} \subseteq \mathbf{f}(\mathbf{x}) \quad (\text{E.8})$$

En général, la fonction d'inclusion n'est pas unique et dépend de la manière dont  $f$  est exprimée. On peut cependant définir la fonction d'inclusion naturelle  $\mathbf{f}$  de  $f$  qui s'obtient en remplaçant chaque variable réelle  $x_i$  par sa variable correspondante  $x_i$  et chaque opération arithmétique par l'opération sur les intervalles équivalente. La fonction d'inclusion est minimale si  $f$  est continue et que chaque variable n'apparaît qu'une seule fois.

### Exemple

On considère la même fonction  $f$  écrite sous quatre formes différentes

$$\begin{aligned} f_1(x) &= x^2 + 2x \\ f_2(x) &= x(x + 2) \\ f_3(x) &= x.x + 2x \\ f_4(x) &= (x + 1)^2 - 1 \end{aligned}$$

L'évaluation des fonctions d'inclusions naturelles de ces quatre expressions sur l'intervalle  $x = [-1, 1]$  donne

$$\begin{aligned} f_1(x) &= x^2 + 2x = [-2, 3] \\ f_2(x) &= x(x + 2) = [-3, 3] \\ f_3(x) &= x \cdot x + 2x = [-3, 3] \\ f_4(x) &= (x + 1)^2 - 1 = [-1, 3] \end{aligned}$$

On remarque que la taille des intervalles, obtenus par le biais de ces quatre fonctions d'inclusion, dépend de l'expression utilisée pour l'écriture de  $f$ . Comme  $f$  est continue, la fonction d'inclusion naturelle minimale est  $f_4$  puisque  $x$  apparaît une seule fois. Ainsi  $f_4$  permet de trouver le plus petit intervalle contenant l'image de  $x$ .  $\square$

Posons un exemple classique issu de [Moore 1966] pour caractériser le phénomène dit d'enveloppement.

### Exemple

On considère des rotations successives d'un pavé  $\mathbf{x} = x_1 \times x_2$  à l'aide d'une rotation définie par la matrice

$$A = \begin{bmatrix} \cos \theta & \sin \theta \\ -\sin \theta & \cos \theta \end{bmatrix} \quad (\text{E.9})$$

Une rotation du pavé  $\mathbf{x}$  donne lieu à un rectangle de même taille, tracé sur la figure E.1. La représentation de ce rectangle par un pavé se fait en l'enveloppant par un autre rectangle dont les cotés sont parallèles aux axes du repère. La rotation suivante de ce pavé dont la taille est plus grande que celle du pavé initial donne lieu à un autre rectangle de même taille. Sa représentation par un pavé augmente sa taille. Par suite, la rotation successive d'un pavé génère une suite de pavés de tailles croissantes alors que la rotation est une opération conservatrice. Afin de contrer cet effet lors de la résolution de  $P_1$  et  $P_2$ , la taille des pavés doit être réduite. En effet, plus la finesse des pavés sera grande, moins le pessimisme sera important.  $\square$

## E.2 Inversion ensembliste par arithmétique d'intervalles

La définition d'une région de l'espace répondant à un ensemble de contraintes est généralement perçue comme un problème de satisfaction de contraintes.

### Définition E.5 (Problème de satisfaction de contraintes [Granvilliers & Benhamou 2007])

Soit un vecteur d'inconnus  $(x_1, \dots, x_n) \in \mathbb{R}^n$ , et soit un système de contraintes  $(c, \mathbf{x})$  défini par un ensemble de contraintes  $c = \{c_1, \dots, c_{n_c}\}$  et un domaine  $\mathbf{x} = x_1 \times \dots \times x_n$ . Nous cherchons un sous ensemble  $\mathbb{S}$  de  $\mathbb{R}^n$  tel que :

$$\begin{cases} c_i : f_i(x_1, \dots, x_n) = 0, & i = 1, \dots, s, \\ c_j : f_j(x_1, \dots, x_n) \leq 0, & j = s + 1, \dots, n_c, \\ x_l \in x_l & l = 1, \dots, n \end{cases} \quad (\text{E.10})$$

Le jeu de contraintes  $c$  est constitué de fonctions généralement non linéaires et non nécessairement différentiables.

Un tel problème peut se réécrire sous la forme d'un problème d'inversion ensembliste.

**Définition E.6 (inversion ensembliste)**

On souhaite résoudre l'équation :

$$f(x) \in \mathbf{y}, \text{ avec } x \in \mathbf{x} \subset \mathbb{R}^n, \mathbf{y} \in \mathbb{IR}^{n_c} \quad (\text{E.11})$$

L'ensemble  $\mathbb{S}$  des solutions de (E.11) est donné par :

$$\mathbb{S} = \{x \in \mathbf{x} | f(x) \in \mathbf{y}\} \quad (\text{E.12})$$

Cet ensemble peut-être réécrit sous la forme suivante :

$$\mathbb{S} = f^{-1}(\mathbf{y}) \cap \mathbf{x} \quad (\text{E.13})$$

La caractérisation de l'ensemble  $\mathbf{x}$ , pour le cas non linéaire, peut être résolu de manière garantie par un algorithme d'inversion ensembliste. Ce type d'algorithme *branch-and-bound* permet, par partitionnement de l'ensemble initial de recherche, de trouver un encadrement lorsqu'au moins une solution existe, de l'ensemble des solutions. On note respectivement par  $\underline{\mathbb{S}}$  et  $\overline{\mathbb{S}}$  l'encadrement intérieur et l'encadrement extérieur de  $\mathbb{S}$  tel que :

$$\underline{\mathbb{S}} \subseteq \mathbb{S} \subseteq \overline{\mathbb{S}} \quad (\text{E.14})$$

Donnons l'exemple de l'algorithme SIVIA [Jaulin & Walter 1993] :

<p>SIVIA(Entrées : <math>\mathbf{x}</math> , <math>\eta</math> ; Sorties : <math>\underline{\mathbb{S}}</math>, <math>\overline{\mathbb{S}}</math>)</p> <ol style="list-style-type: none"> <li>1. Si <math>\mathbf{f}(\mathbf{x}) \cap \mathbf{y} = \emptyset</math>, alors rejeter <math>\mathbf{x}</math></li> <li>2. Si <math>\mathbf{f}(\mathbf{x}) \subseteq \mathbf{y}</math>, alors <math>\underline{\mathbb{S}} := \underline{\mathbb{S}} \cup \mathbf{x}</math>; <math>\overline{\mathbb{S}} := \overline{\mathbb{S}} \cup \mathbf{x}</math></li> <li>3. Si <math>w(\mathbf{x}) \leq \eta</math>, alors <math>\overline{\mathbb{S}} := \overline{\mathbb{S}} \cup \mathbf{x}</math></li> <li>4. bissecter <math>\mathbf{x}</math> en <math>(\mathbf{x}_1, \mathbf{x}_2)</math>;</li> </ol> <p>SIVIA(Entrées : <math>\mathbf{x}_1</math> , <math>\eta</math> ; Sorties : <math>\underline{\mathbb{S}}</math>, <math>\overline{\mathbb{S}}</math>) ; SIVIA(Entrées : <math>\mathbf{x}_2</math> , <math>\eta</math> ; Sorties : <math>\underline{\mathbb{S}}</math>, <math>\overline{\mathbb{S}}</math>)</p>
--

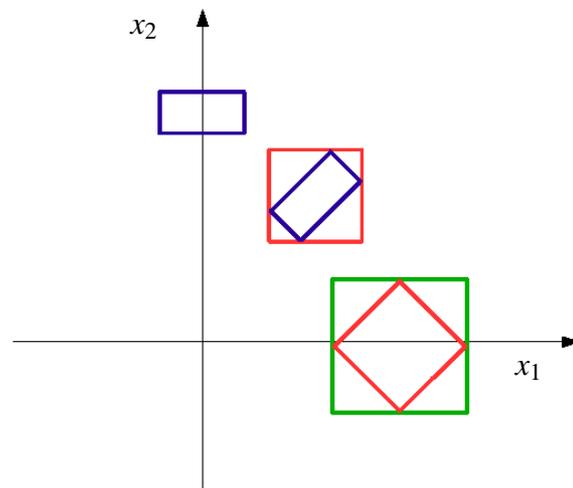
TAB. E.1 – Algorithme SIVIA

L'ensembles des pavés

$$\Delta\mathbb{S} = \overline{\mathbb{S}} \setminus \underline{\mathbb{S}}$$

représente l'incertitude sur la caractérisation de l'ensemble solution  $\mathbb{S}$ , il contient des pavés de tailles plus petites que  $\eta$ . Grâce à l'utilisation de techniques *branch-and-bound*, ces algorithmes sont connus pour être numériquement *complets* i.e. aucune solution n'est perdue [Jaulin & Walter 1993]. Leur complexité est exponentielle avec la dimension du problème à résoudre [Hansen 1992].

Pour améliorer ce type d'algorithme, des contracteurs sont régulièrement utilisés afin de réduire l'ensemble de recherche avant de bissecter [Hansen 1992, Neumaier 1990].

FIG. E.1 – Effets d'enveloppement pour  $\theta = -\pi/4$