



**HAL**  
open science

# Contribution au développement d'une loi de guidage autonome par platitude. Application à une mission de rentrée atmosphérique.

Vincent Morio

## ► To cite this version:

Vincent Morio. Contribution au développement d'une loi de guidage autonome par platitude. Application à une mission de rentrée atmosphérique.. Automatique / Robotique. Université Sciences et Technologies - Bordeaux I, 2009. Français. NNT: . tel-00394728v2

**HAL Id: tel-00394728**

**<https://theses.hal.science/tel-00394728v2>**

Submitted on 14 Jun 2009 (v2), last revised 24 Nov 2018 (v3)

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

N° d'ordre : 3797

# THÈSE

présentée à

**L'UNIVERSITÉ BORDEAUX I**

ÉCOLE DOCTORALE DES SCIENCES PHYSIQUES ET DE L'INGÉNIEUR

par

**Vincent MORIO**

en vue d'obtenir le grade de

**DOCTEUR**

SPÉCIALITÉ : AUTOMATIQUE

---

*Contribution au développement  
d'une loi de guidage autonome par platitude  
Application à une mission de rentrée atmosphérique*

---

*Soutenue le 19 mai 2009*

**Après avis de MM.**

Jean LÉVINE  
Daniel ALAZARD

Professeur, École des Mines de Paris  
Professeur, Supaéro, Toulouse

*Rapporteur*  
*Rapporteur*

**Devant la commission d'examen formée de :**

Pascal FOUILLAT  
Jean LÉVINE  
Daniel ALAZARD  
Gary J. BALAS  
Christian PHILIPPE  
Marie-Catherine PALAU  
Ali ZOLGHADRI  
Franck CAZAURANG

Professeur, Université Bordeaux 1  
Professeur, École des Mines de Paris  
Professeur, Supaéro, Toulouse  
Professeur, Université du Minnesota, USA  
Agence Spatiale Européenne, Pays-Bas  
Astrium Space Transportation  
Professeur, Université Bordeaux 1  
Maître de conférences, Université Bordeaux 1

*Président*  
*Examineur*  
*Examineur*  
*Examineur*  
*Examineur*  
*Examineur*  
*Directeur de thèse*  
*Co-directeur de thèse*

– 2009 –



*À mes parents, mon frère, ma soeur.*





# Remerciements

Ce mémoire présente les travaux de recherche que j'ai effectué depuis octobre 2006 au sein de l'équipe ARIA<sup>1</sup> du laboratoire de l'Intégration du Matériau au Système (IMS), unité mixte de recherche du Centre National de la Recherche Scientifique (CNRS).

Je tiens tout d'abord à remercier chaleureusement mon directeur de thèse, Monsieur Ali Zolghadri, Professeur à l'Université Bordeaux 1, pour m'avoir accueilli au sein de son équipe de recherche, et pour m'avoir accordé toute sa confiance et son soutien durant mes travaux de recherche. Je tiens également à remercier chaleureusement mon co-directeur de thèse, Monsieur Franck Cazaurang, Maître de conférences à l'Université Bordeaux 1, pour avoir soutenu et encouragé l'ensemble de mes travaux de recherche durant ces trois années, pour m'avoir transmis sa passion pour l'Automatique et pour m'avoir prodigué de nombreux conseils. En outre, je tiens à les remercier tous les deux pour la qualité de leur encadrement.

Que Messieurs Jean Lévine, Directeur de Recherches à l'École Nationale Supérieure des Mines de Paris, et Daniel Alazard, Professeur à l'Institut Supérieur de l'Aéronautique et de l'Espace (ISAE), trouvent ici l'expression de ma plus profonde gratitude pour avoir accepté d'expertiser mon manuscrit de thèse et de faire partie de mon jury. Leurs critiques et leurs suggestions ont permis d'améliorer de manière significative la qualité de ce mémoire. Je remercie tout particulièrement Monsieur Jean Lévine, qui m'a accueilli à maintes reprises à l'École Nationale Supérieure des Mines de Paris dans le cadre de réunions de travail sur la théorie des systèmes plats.

Je tiens également à exprimer toute ma reconnaissance à Madame Marie-Catherine Palau, EADS Astrium Space Transportation, Monsieur Gary Balas, Professeur à l'Université du Minnesota, Monsieur Christian Philippe, Agence Spatiale Européenne (ESA) et Monsieur Pascal Fouillat, Professeur à l'Université Bordeaux 1, qui m'ont fait l'honneur de participer à mon jury de thèse.

Parmi les rencontres qui m'ont conduit à entreprendre ces travaux de recherche, je pense tout particulièrement à Monsieur Philippe Vernis, EADS Astrium Space Transportation, qui m'a initié aux techniques de guidage pour les véhicules de rentrée atmosphérique et qui m'a transmis sa passion pour l'aérospatial.

---

<sup>1</sup>Approche Robuste et Intégrée de l'Automatique.

## *Remerciements*

---

Mes remerciements s'adressent également à Messieurs David Henry, Loic Lavigne, Tarek Raissi, Christophe Farges, Jérôme Cieslak, Maîtres de conférences à l'Université Bordeaux 1, pour leur convivialité, leur soutien et leurs précieux conseils. Je tiens également à remercier vivement Monsieur Christophe Louembet, Maître de conférences à l'Université Paul Sabatier, pour les nombreuses discussions que nous avons eues sur la commande optimale et les formes géométriques superquadriques. Je voudrais également remercier mes collègues de bureau, Alexandre Falcoz, Gaétan Videau, Pascal Simon et Irwin Issuri, qui ont contribué à ce que ces trois années de recherche se déroulent dans un cadre très agréable.

Merci également aux stagiaires que j'ai eu la chance de co-encadrer avec Monsieur Franck Cazaurang. Je pense notamment à Mickael Lefebvre et Patrice Lecoq qui m'ont aidé à développer le simulateur de la navette spatiale américaine.

Je ne pourrais clore ces remerciements sans une pensée très personnelle pour les membres de ma famille. Je remercie très chaleureusement mes parents, mon frère et ma soeur pour leur soutien inconditionnel durant ces trois années de recherche. Ce mémoire leur est dédié.

# Acknowledgments

The period as a member of the ARIA<sup>2</sup> group at Bordeaux University has been most enjoyable. I would like to express my sincere gratitude to my supervisors, Prof. Ali Zolghadri and Dr. Franck Cazaurang. Their encouragement and support have been invaluable.

I would like to thank the members of my examination board, and particularly my reviewers, Prof. Jean Lévine and Prof. Daniel Alazard for providing me with highly appreciated comments and tips for improving the quality of my manuscript.

I also would like to thank all members of ARIA group who contribute every day to make a very inspiring teamwork.

Special thanks to Prof. Gary Balas. I met him at San Antonio, Texas, during CACSD conference (2008). I would like to express my gratitude to him for making a long trip to Bordeaux in order to participate to my examination committee.

Finally, I would like to dedicate this manuscript to my parents, my brother and my sister for their invaluable support during my research work.

---

<sup>2</sup>Robust and Integrated Approach to Automatic Control.

## *Acknowledgments*

---

# Table des matières

Glossaire	15
Table des figures	21
Liste des tableaux	25
Liste des algorithmes	27
Liste des publications	29
Introduction générale	31
<b>Partie I Caractérisation formelle de sorties plates pour les systèmes non linéaires</b>	<b>37</b>
Position du problème	39
<b>Chapitre 1 Platitude différentielle</b>	<b>41</b>
1.1 Introduction . . . . .	42
1.2 Rappels sur le concept de platitude . . . . .	42
1.3 Conditions nécessaires et suffisantes de platitude . . . . .	49
1.4 Conclusion . . . . .	56
<b>Chapitre 2 Considérations pratiques sur quelques algorithmes de calcul des sorties plates</b>	<b>57</b>
2.1 Introduction . . . . .	58
2.2 Calcul de sorties plates pour les systèmes linéaires à retards . . . . .	58
2.2.1 Résultat principal . . . . .	59

2.2.2 Exemples . . . . .	60
2.3 Conclusion . . . . .	67
<b>Chapitre 3 Application à un véhicule de rentrée atmosphérique</b>	<b>69</b>
3.1 Introduction . . . . .	70
3.2 Platitude du modèle “Terre plate” . . . . .	70
3.3 Platitude du modèle “Terre tournante” . . . . .	75
3.4 Conclusion . . . . .	81

**Partie II Planification autonome de trajectoires de rentrée atmosphérique** **83**

**Position du problème** **85**

<b>Chapitre 1 État de l’art</b>	<b>87</b>
1.1 Introduction . . . . .	88
1.2 Programmation non linéaire . . . . .	88
1.2.1 Méthode de Newton . . . . .	88
1.2.2 Problème d’optimisation sans contraintes . . . . .	89
1.2.3 optimisation sous contraintes égalitaires . . . . .	91
1.2.4 Optimisation sous contraintes inégalitaires . . . . .	93
1.2.5 Programmation non linéaire . . . . .	94
1.3 Commande optimale . . . . .	101
1.3.1 Contraintes dynamiques . . . . .	102
1.3.2 Contraintes égalitaires algébriques . . . . .	103
1.3.3 Contraintes inégalitaires algébriques . . . . .	105
1.4 Propagation de trajectoires . . . . .	105
1.5 Méthodes classiques d’optimisation de trajectoires . . . . .	109
1.5.1 Méthodes directes de tirs . . . . .	110
1.5.2 Méthodes indirectes de tirs . . . . .	116
1.5.3 Méthodes de tirs multiples . . . . .	121
1.5.4 Méthodes de transcription indirecte . . . . .	127
1.5.5 Méthodes de transcription directe . . . . .	130
1.5.6 Quelques autres méthodes . . . . .	135

---

1.6	Conclusion . . . . .	137
<b>Chapitre 2 Méthodologie proposée</b>		<b>141</b>
2.1	Introduction . . . . .	142
2.2	Architecture générale de la loi de guidage autonome . . . . .	143
2.3	Planification de trajectoires par platitude . . . . .	149
2.3.1	Classe de problèmes de commande optimale considérée . . . . .	149
2.3.2	Rappels sur le concept de platitude différentielle . . . . .	150
2.3.3	Problème de commande optimale dans l’espace des sorties plates	151
2.4	Convexification du problème de commande optimale . . . . .	154
2.4.1	Formes superquadriques généralisées . . . . .	155
2.4.2	Problème d’annexion superellipsoïdale . . . . .	180
2.4.3	Résolution par algorithme génétique . . . . .	181
2.4.4	Problème de commande optimale convexe . . . . .	186
2.4.5	Résultats préliminaires . . . . .	188
2.5	Planification de trajectoires tolérante aux défauts . . . . .	189
2.5.1	Cartes de trimmabilité statique . . . . .	192
2.5.2	Première solution: prise en compte directe des contraintes d’équi- librage dans le problème de commande optimale . . . . .	199
2.5.3	Seconde solution: replanification “pilotée” de trajectoires . . . . .	202
2.6	Transformation en un problème de programmation non linéaire . . . . .	205
2.7	Conclusion . . . . .	210
<b>Chapitre 3 Application à une mission de rentrée atmosphérique</b>		<b>213</b>
3.1	Introduction . . . . .	214
3.2	Problématique associée au guidage pour les phases TAEM et A&L . . . . .	214
3.2.1	Navette spatiale américaine . . . . .	215
3.2.2	Phases de dissipation d’énergie en zone terminale et d’atterris- sage automatique . . . . .	219
3.3	Planification de trajectoires de rentrée atmosphérique par platitude . . . . .	228
3.3.1	Modélisation des coefficients aérodynamiques . . . . .	228
3.3.2	Planification de trajectoires TAEM par platitude . . . . .	234
3.3.3	Planification de trajectoires A&L par platitude . . . . .	250
3.4	Planification de trajectoires de rentrée en présence de perturbations aérologiques . . . . .	256



3.4.1	Champs de cisaillement de vent considérés . . . . .	258
3.4.2	Intégration des perturbations aérologiques dans le processus de planification de trajectoires . . . . .	258
3.5	Convexification du problème de commande optimale . . . . .	266
3.5.1	Approximation convexe intérieure des contraintes de trajectoire	267
3.5.2	Problème de commande optimale convexe . . . . .	271
3.6	Prise en compte de défaillances de gouvernes . . . . .	276
3.6.1	Expressions des coefficients de forces et de moments aérodyna- miques de l'orbiteur STS-1 . . . . .	276
3.6.2	Génération des cartes de trimmabilité statique . . . . .	282
3.6.3	Replanification de trajectoires TAEM en présence de défauts . .	283
3.7	Conclusion . . . . .	290
<b>Conclusion générale</b>		<b>293</b>
<b>Annexes</b>		<b>297</b>
<b>Annexe A Forme normale de Smith</b>		<b>297</b>
<b>Annexe B Fonctions polynomiales par morceaux et B-splines</b>		<b>301</b>
<b>Annexe C Équations dynamiques associées à un véhicule de rentrée atmosphérique</b>		<b>305</b>
C.1	Trièdres de référence . . . . .	305
C.1.1	Trièdres inertiel et géocentrique . . . . .	305
C.1.2	Trièdre pointant vers le véhicule . . . . .	305
C.1.3	Trièdre cinématique . . . . .	306
C.1.4	Trièdre aérodynamique . . . . .	307
C.1.5	Trièdre stabilité . . . . .	307
C.1.6	Trièdre véhicule . . . . .	308
C.2	Équations dynamiques du véhicule . . . . .	309
C.2.1	Équations de position . . . . .	309
C.2.2	Équations de vitesse . . . . .	311
C.2.3	Équations de vitesses angulaires . . . . .	314
C.2.4	Équations d'attitude . . . . .	316
C.2.5	Équations additionnelles . . . . .	319

---

<b>Bibliographie</b>	<b>321</b>
<b>Index</b>	<b>333</b>



# Glossaire

## Notations

$\alpha$	Angle d'incidence
$\beta$	Angle de dérapage
$\mu$	Angle de gîte aérodynamique
$\Omega$	Vitesse de rotation terrestre
$x$	Portée longitudinale
$y$	Portée latérale
$h$	Altitude
$\varphi$	Latitude
$\theta$	Longitude
$R$	Rayon-vecteur
$V$	Vitesse relative
$M$	Nombre de Mach
$E$	Energie totale
$\gamma$	Pente sur la trajectoire
$\chi$	Cap
$g$	Accélération gravitationnelle
$\rho$	Densité atmosphérique
$\rho_0$	Densité atmosphérique au niveau moyen de la mer
$m$	Masse du véhicule
$S$	Surface aérodynamique de référence
$H_{ref}$	Facteur d'échelle du modèle de densité atmosphérique exponentiel
$\bar{q}$	Pression dynamique
$\Phi$	Flux thermique au point d'arrêt
$\Gamma$	Facteur de charge
$\mu_T$	Constante de gravitation terrestre
$R_T$	Rayon terrestre moyen
$b$	Envergure
$\bar{c}$	Longueur de la corde moyenne du profil de l'aile
$\tau$	Temps normalisé
$V_s$	Vitesse moyenne du son
$L$	Force de portance
$Y$	Force latérale
$D$	Force de traînée

$I_{xx}$	Moment d'inertie sur l'axe $x$
$I_{yy}$	Moment d'inertie sur l'axe $y$
$I_{zz}$	Moment d'inertie sur l'axe $z$
$I_{xy}$	Produit d'inertie sur les axes $x$ et $y$
$I_{xz}$	Produit d'inertie sur les axes $x$ et $z$
$I_{yz}$	Produit d'inertie sur les axes $y$ et $z$
$x_{mrc}$	Position du centre instantané de rotation sur l'axe $x$
$y_{mrc}$	Position du centre instantané de rotation sur l'axe $y$
$z_{mrc}$	Position du centre instantané de rotation sur l'axe $z$
$x_{cg}$	Position du centre de gravité sur l'axe $x$
$y_{cg}$	Position du centre de gravité sur l'axe $y$
$z_{cg}$	Position du centre de gravité sur l'axe $z$
$C_{L_0}$	Coefficient de portance en configuration lisse
$C_{D_0}$	Coefficient de traînée en configuration lisse
$C_{Y_0}$	Coefficient de force latérale en configuration lisse
$C_{l_0}$	Coefficient de moment de roulis en configuration lisse
$C_{m_0}$	Coefficient de moment de tangage en configuration lisse
$C_{n_0}$	Coefficient de moment de lacet en configuration lisse
$C_{L\delta_e}$	Contribution des élevons $\delta_e$ sur le coefficient de portance
$C_{D\delta_e}$	Contribution des élevons $\delta_e$ sur le coefficient de traînée
$C_{Y\delta_e}$	Contribution des élevons $\delta_e$ sur le coefficient de force latérale
$C_{l\delta_e}$	Contribution des élevons $\delta_e$ sur le coefficient de moment de roulis
$C_{m\delta_e}$	Contribution des élevons $\delta_e$ sur le coefficient de moment de tangage
$C_{n\delta_e}$	Contribution des élevons $\delta_e$ sur le coefficient de moment de lacet
$C_{Y\delta_a}$	Contribution des ailerons $\delta_a$ sur le coefficient de force latérale
$C_{l\delta_a}$	Contribution des ailerons $\delta_a$ sur le coefficient de moment de roulis
$C_{n\delta_a}$	Contribution des ailerons $\delta_a$ sur le coefficient de moment de lacet
$C_{L\delta_{sb}}$	Contribution de l'aérofrein $\delta_{sb}$ sur le coefficient de portance
$C_{D\delta_{sb}}$	Contribution de l'aérofrein $\delta_{sb}$ sur le coefficient de traînée
$C_{Y\delta_{sb}}$	Contribution de l'aérofrein $\delta_{sb}$ sur le coefficient de force latérale
$C_{l\delta_{sb}}$	Contribution de l'aérofrein $\delta_{sb}$ sur le coefficient de moment de roulis
$C_{m\delta_{sb}}$	Contribution de l'aérofrein $\delta_{sb}$ sur le coefficient de moment de tangage
$C_{n\delta_{sb}}$	Contribution de l'aérofrein $\delta_{sb}$ sur le coefficient de moment de lacet
$C_{Y\delta_r}$	Contribution de la gouverne de direction $\delta_r$ sur le coeff. de force latérale
$C_{l\delta_r}$	Contribution de la gouverne de direction $\delta_r$ sur le coeff. de moment de roulis
$C_{n\delta_r}$	Contribution de la gouverne de direction $\delta_r$ sur le coeff. de moment de lacet
$C_{L\delta_{bf}}$	Contribution du body flap $\delta_{bf}$ sur le coefficient de portance
$C_{D\delta_{bf}}$	Contribution du body flap $\delta_{bf}$ sur le coefficient de traînée
$C_{Y\delta_{bf}}$	Contribution du body flap $\delta_{bf}$ sur le coefficient de force latérale
$C_{l\delta_{bf}}$	Contribution du body flap $\delta_{bf}$ sur le coefficient de moment de roulis
$C_{m\delta_{bf}}$	Contribution du body flap $\delta_{bf}$ sur le coefficient de moment de tangage
$C_{n\delta_{bf}}$	Contribution du body flap $\delta_{bf}$ sur le coefficient de moment de lacet
$C_{l_p}$	Effet de la vitesse de roulis $p$ sur le coefficient de moment de roulis

---

$C_{l\dot{r}}$	Effet de la vitesse de lacet $r$ sur le coefficient de moment de roulis
$C_{m\dot{q}}$	Effet de la vitesse de tangage $q$ sur le coefficient de moment de tangage
$\Delta C_{m\dot{q}}$	Effets couplés de l'aérofrein $\delta_{sb}$ , du body flap $\delta_{bf}$ et de la vitesse de tangage $q$ sur le coefficient de moment de tangage
$C_{n\dot{p}}$	Effet de la vitesse de roulis $p$ sur le coefficient de moment de lacet
$C_{n\dot{r}}$	Effet de la vitesse de lacet $r$ sur le coefficient de moment de lacet
$K_{L\delta_e}$	Coefficient d'efficacité des élévons $\delta_e$ sur le coefficient de portance
$K_{Y\delta_e}$	Coefficient d'efficacité des élévons $\delta_e$ sur le coefficient de force latérale
$K_{l\delta_e}$	Coefficient d'efficacité des élévons $\delta_e$ sur le coefficient de moment de roulis
$K_{m\delta_e}$	Coefficient d'efficacité des élévons $\delta_e$ sur le coeff. de moment de tangage
$K_{m\delta_{sb}}$	Coefficient d'efficacité de l'aérofrein $\delta_{sb}$ sur le coeff. de moment de tangage
$K_{n\delta_e}$	Coefficient d'efficacité des élévons $\delta_e$ sur le coefficient de moment de lacet
$\delta$	Vecteur des braquages de gouvernes
$\delta_e$	Braquage moyen total des élévons
$\delta_{e_{i_l}}$	Braquage de l'élévon intérieur gauche
$\delta_{e_{i_r}}$	Braquage de l'élévon intérieur droit
$\delta_{e_{o_l}}$	Braquage de l'élévon extérieur gauche
$\delta_{e_{o_r}}$	Braquage de l'élévon extérieur droit
$\delta_{e_i}$	Braquage moyen des élévons intérieurs
$\delta_{e_o}$	Braquage moyen des élévons extérieurs
$\delta_{e_l}$	Braquage moyen des élévons gauches
$\delta_{e_r}$	Braquage moyen des élévons droits
$\delta_a$	Braquage moyen total des ailerons
$\delta_{a_i}$	Braquage moyen des ailerons intérieurs
$\delta_{a_o}$	Braquage moyen des ailerons extérieurs
$\delta_r$	Braquage de la gouverne de direction
$\delta_{bf}$	Braquage du body flap
$\delta_{sb}$	Braquage de l'aérofrein

## Acronymes

ALI	Auto-Landing Interface ( <i>interface avec la phase d'atterrissage automatique</i> )
AMS	Attainable Moment Subset ( <i>sous-ensemble des moments atteignables</i> )
ARD	Atmospheric Reentry Demonstrator ( <i>démonstrateur de rentrée atmosphérique</i> )
ARES	Atmospheric Reentry Experimental Spaceplane ( <i>avion spatial expérimental de rentrée atmosphérique</i> )
A&L	Auto-Landing ( <i>atterrissage automatique</i> )
AG&C	Advanced Guidance and Control ( <i>techniques de guidage et de pilotage avancées</i> )
ANN	Analytical Neural Network ( <i>réseau de neurones analytique</i> )
ASTOS	AeroSpace Trajectory Optimization Software ( <i>logiciel d'optimisation de trajectoires pour applications aérospatiales</i> )
CAD	Computer-Aided Design ( <i>conception assistée par ordinateur</i> )

DAE	Differential Algebraic Equation ( <i>équation algébrique différentielle</i> )
ET	External Tank ( <i>réservoir externe</i> )
FDI	Fault Detection and Isolation ( <i>détection et localisation de défauts</i> )
FDIR	Fault Detection, Isolation and Reconfiguration ( <i>détection, isolation de défauts et reconfiguration</i> )
FL	Fuzzy Logic ( <i>logique floue</i> )
FTOPP	Fault-Tolerant Onboard Path Planner ( <i>planificateur de trajectoires en ligne tolérant aux défauts</i> )
GA	Genetic Algorithm ( <i>algorithme génétique</i> )
G&C	Guidance and Control ( <i>guidage et pilotage</i> )
GNC	Guidance, Navigation and Control ( <i>guidage, navigation et pilotage</i> )
HAC	Heading Alignment Cylinder (or Cone) ( <i>cylindre (ou cône) d'alignement</i> )
ISS	International Space Station ( <i>station spatiale internationale</i> )
IVP	Initial Value Problem ( <i>problème aux valeurs initiales</i> )
IXV	Intermediate eXperimental Vehicle ( <i>véhicule expérimental intermédiaire</i> )
KKT	Karush-Kuhn-Tucker
L/D	Lift-over-Drag ratio ( <i> finesse</i> )
LFT	Linear Fractional Transformation ( <i>transformation linéaire fractionnaire</i> )
LMI	Linear Matrix Inequality ( <i>inégalité matricielle linéaire</i> )
LMSE	Learning Mean Square Error ( <i>erreur quadratique moyenne sur l'ensemble d'apprentissage</i> )
LP	Linear Programming ( <i>programmation linéaire</i> )
LPV	Linear Parameter Varying ( <i>linéaire à paramètres variants</i> )
LTV	Linear Time Varying ( <i>linéaire à temps variant</i> )
MILP	Mixed-Integer Linear Programming ( <i>programmation linéaire entier mixte</i> )
MLP	Multi-Layer Perceptron ( <i>perceptron multi-couches</i> )
NDI	Nonlinear Dynamic Inversion ( <i>inversion dynamique non linéaire</i> )
NLP	Nonlinear Programming ( <i>programmation non linéaire</i> )
OCEANS	Optimal Convexification by Evolutionnary Algorithm aNd Superquadrics ( <i>convexification optimale par algorithme évolutionniste et superquadriques</i> )
OCP	Optimal Control Problem ( <i>problème de commande optimale</i> )
ODE	Ordinary Differential Equation ( <i>équation différentielle ordinaire</i> )
OMS	Orbital Maneuvering System ( <i>système de manoeuvres orbitales</i> )
PDE	Partial Differential Equation ( <i>équation aux dérivées partielles</i> )
PID	Principal Ideal Domain ( <i>anneau principal</i> )
QP	Quadratic Programming ( <i>programmation quadratique</i> )
RCS	Reaction Control System ( <i>système de contrôle d'attitude</i> )
SBX	Simulated Binary Crossover ( <i>croisement binaire simulé</i> )
SDP	Semi-Definite Programming ( <i>programmation semi-définie</i> )
SQP	Sequential Quadratic Programming ( <i>programmation quadratique séquentielle</i> )
SRB	Solid Rocket Booster ( <i>propulseur d'appoint à poudre</i> )
SSME	Space Shuttle Main Engine ( <i>moteur principal</i> )
STS	Space Transportation System ( <i>système de transport spatial</i> )
SVD	Singular Value Decomposition ( <i>décomposition en valeurs singulières</i> )
TAEM	Terminal Area Energy Management ( <i>dissipation d'énergie en zone terminale</i> )

---

TEP	TAEM Entry Point ( <i>point d'entrée TAEM</i> )
TMSE	Training Mean Square Error ( <i>erreur quadratique moyenne sur l'ensemble de validation</i> )





# Table des figures

1.1	véhicule roulant sans glisser sur un plan. . . . .	43
1.2	système masses-ressorts. . . . .	44
1.3	corde vibrante avec une masse. . . . .	48
1.1	illustration de la phase d'atterrissage automatique. . . . .	113
1.2	processus de planification de trajectoires d'atterrissage. . . . .	115
1.3	effet de la variation de $x_0$ sur la trajectoire d'atterrissage. . . . .	116
1.4	description géométrique des trajectoires TAEM et A&L de référence. . . . .	119
1.5	segmentation de la trajectoire de référence TAEM. . . . .	124
2.1	architecture de la loi de guidage autonome. . . . .	144
2.2	architecture de la loi de pilotage non linéaire. . . . .	147
2.3	organigramme illustrant la méthodologie de planification de trajectoires proposée. . . . .	148
2.4	équivalence entre les trajectoires du système initial et celles du système trivial. . . . .	152
2.5	illustration géométrique des paramètres de dimensionnement d'un superel- lisoïde 3D. . . . .	158
2.6	exemples de formes superquadriques 3D. . . . .	159
2.7	paramétrisation trigonométrique explicite d'un superellipsoïde 3D. . . . .	162
2.8	paramétrisation angle-centre explicite d'un superellipsoïde 3D. . . . .	165
2.9	effet d'une rotation sur une forme superquadrique 3D. . . . .	167
2.10	effet d'un pincement linéaire sur une forme superquadrique 3D. . . . .	168
2.11	contours d'intégration considérés pour déterminer l'aire une superellipse. . . . .	173
2.12	interprétation d'un superellipsoïde 3D comme un empilement de superel- lipses d'épaisseurs infinitésimales $dz$ . . . . .	174
2.13	organigramme d'un algorithme génétique classique. . . . .	183
2.14	résolution du problème d'annexion superellipsoïdale par algorithme gé- nétique. . . . .	184
2.15	illustration du processus d'annexion superellipsoïdale. . . . .	190
2.16	exemple de carte d'équilibrage statique. . . . .	195
2.17	autre exemple de carte d'équilibrage statique. . . . .	196
2.18	illustration du coefficient de portance en situation nominale et défailante. . . . .	197
2.19	illustration du coefficient de traînée en situation nominale et défailante. . . . .	198

2.20	B-spline comportant 6 intervalles ( $l = 6$ ), d'ordre 4 ( $k = 4$ ), et de multiplicité 3 ( $s = 3$ ). Les contraintes sont satisfaites en 21 points de collocation régulièrement espacés. Les neuf points de contrôle représentent alors les variables de décision du problème. . . . .	207
3.1	illustration du système de lancement Shuttle (STS). . . . .	216
3.2	description de l'orbiteur. . . . .	217
3.3	illustration d'une mission de rentrée atmosphérique. . . . .	221
3.4	phase de dissipation d'énergie en zone terminale (TAEM). . . . .	222
3.5	phase d'atterrissage automatique (A&L). . . . .	226
3.6	vecteurs singuliers à gauche et à droite associés à la plus grande valeur singulière de $A_{C_D}$ . . . . .	231
3.7	erreur d'approximation du coefficient de traînée en configuration lisse. . . . .	233
3.8	contribution de la gouverne de direction sur le coefficient de moment de lacet. . . . .	234
3.9	coefficient de portance tabulé en configuration lisse. . . . .	237
3.10	discrétisation des conditions cinématiques initiales de position. . . . .	240
3.11	trajectoires TAEM en présence de dispersions cinématiques initiales. . . . .	240
3.12	densité de probabilité associée à la durée des trajectoires TAEM obtenues. . . . .	241
3.13	profils des états de référence. . . . .	242
3.14	profils des consignes de guidage de référence. . . . .	243
3.15	profils des contraintes mécaniques le long de la trajectoire de référence. . . . .	243
3.16	(a) contrainte liée à l'équation autonome du système ( $m/s^2$ ) (b) temps CPU nécessaire pour effectuer les différentes optimisations. . . . .	244
3.17	trajectoire TAEM obtenue sous l'environnement ASTOS. . . . .	245
3.18	profils des commandes de référence obtenus sous l'environnement ASTOS. . . . .	245
3.19	contraintes mécaniques obtenues sous l'environnement ASTOS. . . . .	246
3.20	trajectoire TAEM nominale. . . . .	250
3.21	profils des consignes de guidage de référence. . . . .	251
3.22	contrainte égalitaire de trajectoire ( $N$ ). . . . .	251
3.23	contraintes mécaniques le long de la trajectoire TAEM de référence. . . . .	252
3.24	profil nominal d'incidence pour la condition initiale $x_0=-10$ km. . . . .	255
3.25	trajectoire A&L nominale. . . . .	255
3.26	états de référence sur la trajectoire A&L. . . . .	256
3.27	(a) profil nominal d'incidence pour la condition initiale $x_0=-20$ km (b) contrainte égalitaire de trajectoire ( $m/s$ ). Les points de collocation sont représentés par des croix rouges. . . . .	257
3.28	contraintes mécaniques le long de la trajectoire A&L de référence. . . . .	257
3.29	prise en compte de perturbations aérologiques dans le processus de planification de trajectoires. . . . .	263
3.30	prise en compte de perturbations aérologiques dans le processus de planification de trajectoires. . . . .	264
3.31	(a) vitesse de cisaillement de vent longitudinale (b) vitesse de cisaillement de vent latérale (c) vitesse de cisaillement de vent verticale. . . . .	265
3.32	profils des consignes de guidage avec et sans perturbations aérologiques. . . . .	266

---

3.33	contrainte égalitaire de trajectoire avec et sans perturbations aérologiques ( $N$ ). . . . .	266
3.34	profils des contraintes mécaniques avec et sans perturbations aérologiques.	267
3.35	considérations géométriques sur la contrainte de pression dynamique. . . . .	268
3.36	évolution de l'aptitude des individus au fil des générations. . . . .	269
3.37	approximation de la contrainte de pression dynamique par une forme convexe. Le domaine admissible associé à la contrainte est représenté par un nuage de points de couleur verte. Les points associés à la paramétrisation angle-centre sont représentés par des croix de couleur jaune. . . . .	270
3.38	approximation du domaine de gîte admissible par des formes superellipsoïdales. . . . .	271
3.39	trajectoire TAEM obtenue par résolution du problème convexe. . . . .	272
3.40	profils des consignes de guidage le long de la trajectoire TAEM de référence.	273
3.41	contrainte égalitaire de trajectoire ( $m/s^2$ ). . . . .	273
3.42	contraintes mécaniques le long de la trajectoire TAEM. . . . .	274
3.43	trajectoire TAEM projetée dans le superellipsoïde convexe de pression dynamique. . . . .	274
3.44	trajectoire TAEM projetée dans le superellipsoïde convexe de gîte aérodynamique. . . . .	275
3.45	(a) fonction d'appartenance associée à la contrainte de pression dynamique (b) fonction d'appartenance associée à la contrainte de gîte. . . . .	275
3.46	(a) effet des ailerons sur le coefficient de force latérale (b) effet des ailerons sur le coefficient de moment de roulis (c) effet des ailerons sur le coefficient de moment de lacet. . . . .	280
3.47	(a) effet des élevons sur le coefficient de portance (b) effet du body flap sur le coefficient de portance (c) effet de l'aérofren sur le coefficient de portance.	281
3.48	exemples de cartes d'équilibrage statique pour plusieurs scénarii défaillants.	283
3.49	prise en compte des contraintes d'équilibrage statique pour la planification de trajectoires TAEM. . . . .	286
3.50	profils des consignes de guidage sur la trajectoire TAEM de référence. . . . .	287
3.51	contrainte égalitaire le long de la trajectoire TAEM ( $m/s^2$ ). . . . .	287
3.52	contraintes mécaniques le long de la trajectoire TAEM de référence. . . . .	288
3.53	(a) équilibrage statique en tangage le long de la trajectoire TAEM (b) équilibrage statique en roulis et lacet le long de la trajectoire TAEM. . . . .	288
3.54	(a) braquages de l'élevon intérieur gauche $\delta_{e_{i_l}}$ (b) braquages de l'élevon intérieur droit $\delta_{e_{i_r}}$ (c) braquages de l'élevon extérieur gauche $\delta_{e_{o_l}}$ (d) braquages de l'élevon extérieur droit $\delta_{e_{o_r}}$ . . . . .	289
3.55	(a) braquages de l'aérofren $\delta_{sb}$ (b) braquages du body flap $\delta_{bf}$ . . . . .	290
3.56	projection de la trajectoire TAEM de référence sur la carte d'équilibrage statique. . . . .	291
B.1	B-spline $y(t) \in \mathcal{P}(6, 4, 3)$ comportant 6 intervalles ( $l = 6$ ), d'ordre 4 ( $k = 4$ ), et de multiplicité 3 ( $s = 3$ ). Les neuf points de contrôle représentent alors les variables de décision. . . . .	303

*Table des figures*

---

C.1	trièdres inertiel et géocentrique. . . . .	306
C.2	trièdre pointant vers le véhicule. . . . .	307
C.3	trièdre cinématique. . . . .	308
C.4	trièdre aérodynamique. . . . .	309
C.5	trièdre stabilité. . . . .	310
C.6	trièdre véhicule. . . . .	311

# Liste des tableaux

1.1	paramètres de dimensionnement de la trajectoire d'atterrissage. . . . .	113
1.2	paramètres de dimensionnement des polynômes quadratiques de référence. . . . .	126
1.3	classification des principales techniques d'optimisation de trajectoires de rentrée atmosphérique. . . . .	138
2.1	niveaux de criticité associés à la replanification de trajectoires à bord. . . . .	199
3.1	limites et vitesses de braquage des gouvernes. . . . .	218
3.2	caractéristiques principales de la navette spatiale américaine STS-1. . . . .	219
3.3	contraintes de guidage associées à la phase TAEM. . . . .	223
3.4	contraintes associées aux consignes de guidage. . . . .	223
3.5	contraintes de guidage associées à la phase A&L. . . . .	227
3.6	conditions initiales considérées pour l'optimisation de trajectoires TAEM. . . . .	239



# Liste des algorithmes

1	Calcul formel d'une sortie plate. . . . .	52
2	Caractérisation algébrique d'un système linéaire $\delta$ -plat. . . . .	60
3	Algorithme de programmation quadratique (QP). . . . .	95
4	Méthode directe de tirs. . . . .	111
5	Méthode indirecte de tirs. . . . .	117
6	Méthode de tirs multiples. . . . .	123
7	Planification de trajectoires TAEM. . . . .	126
8	Méthode de transcription indirecte. . . . .	129
9	Méthode de transcription directe. . . . .	133
10	Croisement binaire simulé (SBX). . . . .	186
11	Mutation paramétrique. . . . .	187
12	Génération d'une carte d'équilibrage statique. . . . .	198
13	Décomposition de Smith. . . . .	300





# Liste des publications

## Articles dans des revues internationales

- [1] V. Morio, F. Cazaurang and P. Vernis, “Flatness-based Hypersonic Reentry Guidance of a Lifting-body Vehicle”, *Control Engineering Practice*, 17(5) :588-596, May 2009.
- [2] V. Morio, F. Cazaurang, A. Falcoz and P. Vernis, “Robust Terminal Area Energy Management Guidance using Flatness Approach”, *IET Control Theory & Applications*, 2009, to appear.
- [3] V. Morio, F. Cazaurang, A. Zolghadri and P. Vernis, “Onboard Path Planning for Reusable Launch Vehicles. Application to the Shuttle Orbiter Reentry Mission”, *International Review of Aerospace Engineering*, 1(6), December 2008.
- [4] F. Cazaurang, V. Morio, A. Falcoz, D. Henry and A. Zolghadri, “New Model-Based Strategies for Guidance and Health Monitoring of Experimental Reentry Vehicles”, *International Review of Aerospace Engineering*, 1(5) :458-463, October 2008.

## Communications dans des congrès internationaux avec actes et comité de lecture

- [1] V. Morio, F. Cazaurang, A. Zolghadri and P. Vernis, “A new Path Planner based on Flatness Approach. Application to an Atmospheric Reentry Mission”, *Proceedings of the European Control Conference (ECC'09)*, Budapest, Hungary. 2009.
- [2] V. Morio, F. Cazaurang and A. Zolghadri, “On the Formal Characterization of Reduced-Order Flat Outputs over an Ore Algebra”, *Proceedings of the 2nd IEEE Multi-conference on Systems and Control (MSC) / 9th IEEE International Symposium on Computer-aided Control System Design (CACSD)*, pp. 207-214, San Antonio, Texas. 2008.
- [3] V. Morio, F. Cazaurang and A. Zolghadri, “An Effective Algorithm for Analytical Computation of Flat Outputs over the Weyl Algebra”, *Proceedings of the 17th IFAC World Congress*, Seoul, Korea. 2008.
- [4] V. Morio, F. Cazaurang and A. Zolghadri, “Sur la Caractérisation Formelle de Sorties Plates d’Ordre Réduit sur un Algèbre de Weyl”, *Actes de la Conférence Internationale Francophone d’Automatique*, Bucarest, Roumanie. 2008.
- [5] V. Morio, F. Cazaurang, A. Zolghadri and P. Vernis, “Onboard Terminal Area Management Path Planning using Flatness Approach. Application to Shuttle orbiter STS-1 Vehicle”, *Proceedings of the 2nd International ARA Days*, Arcachon, France. 2008.


- [6] V. Morio, A. Falcoz, F. Cazaurang, D. Henry, A. Zolghadri, M. Ganet, P. Vernis and E. Bornschlegl, “SICVER Project : Innovative FDIR Strategies for Experimental Reentry Vehicles”, *Proceedings of the 2nd International ARA Days*, Arcachon, France. 2008.
- [7] V. Morio, A. Falcoz, P. Vernis and F. Cazaurang, “On the design of a flatness-based guidance algorithm for the terminal area energy management of a winged-body vehicle”, *Proceedings of the 17th IFAC Symposium on Automatic Control in Aerospace*, Toulouse, France. 2007.
- [8] P. Vernis, V. Morio and E. Ferreira, “Genetic Algorithm for coupled RLV trajectory and guidance optimization”, *Proceedings of the 17th IFAC Symposium on Automatic Control in Aerospace*, Toulouse, France. 2007.
- [9] V. Morio, P. Vernis and F. Cazaurang, “Hypersonic Reentry and Flatness Theory. Application to medium L/D Entry Vehicle”, *Proceedings of the 1st International ARA Days*, Arcachon, France. 2006.

# Introduction générale

“Il faut toujours viser la lune, car même en cas d’échec,  
on atterrit au milieu des étoiles.”

Oscar Wilde

## Contexte de l’étude

a rentrée atmosphérique et l’exploration planétaire constituent un des axes scientifiques et pluridisciplinaires majeurs à maîtriser pour l’accès à l’espace. Les États-Unis d’Amérique et l’Union Soviétique ont été les premières nations à investir massivement pour parvenir à maîtriser ce problème complexe, et ont développé et expérimenté de nombreuses techniques pour la rentrée atmosphérique. En Europe, un effort important a été déployé depuis une vingtaine d’années. Cet effort a été couronné de succès par la rentrée hypersonique de la capsule ARD (Atmospheric Reentry Demonstrator) dans l’atmosphère terrestre en 1998, et par la sonde Huygens qui s’est posée sur Titan en janvier 2005<sup>3</sup>.

Depuis les premiers balbutiements de l’aéronautique, le concept d’un lanceur réutilisable revient de façon récurrente. L’une des principales difficultés liées à la réutilisation réside dans la dissipation de l’énergie lors du retour d’orbite. En effet, les sous-systèmes requis sont très complexes, chers, et grèvent fortement la fiabilité de la mission, voire la sécurité dans le cas d’une mission habitée. Par ailleurs, les éléments liés à la récupération des étages propulsifs, ailes, surfaces mobiles et trains d’atterrissage, sont lourds et complexes et conditionnent fortement la performance des systèmes réutilisables<sup>4</sup>.

---

<sup>3</sup>après une croisière spatiale de plus de sept ans.

<sup>4</sup>après près de trente ans de service, la navette spatiale américaine, seul système partiellement réutilisable volant actuellement, effectuera son dernier vol dans l’espace le 31 mai 2010, avant sa mise à la retraite et après avoir été l’élément-clé de la construction de la Station Spatiale Internationale (ISS).

Les nouveaux concepts de véhicules, capables d'atteindre une orbite basse avec un seul étage propulsif, permettraient de diminuer considérablement les coûts d'accès à l'espace, tout en augmentant leur fréquence. Le développement de ces lanceurs réutilisables de nouvelle génération s'accompagne de la maîtrise de l'autonomie à bord en dépit d'un nombre réduit de systèmes embarqués. Depuis une dizaine d'années, un effort de recherche important a été mené afin de remplacer, autant que possible, la flexibilité de l'archétype du régulateur autonome, c'est à dire le pilote humain. En effet, afin de garantir une meilleure fiabilité des vols, il est important de pouvoir réduire la charge de travail du pilote durant les phases critiques d'une mission de rentrée atmosphérique, exigeant d'effectuer des tâches complexes en un temps réduit.

De manière générale, l'autonomie d'un véhicule spatial peut être définie par son aptitude à réagir à des modifications de contexte pouvant affecter les objectifs de la mission, et à prendre des décisions correctives appropriées, sans intervention extérieure. Assurer un certain niveau d'autonomie de base, dans le sens d'une automatisation à bord de certaines tâches (exécution de procédures pré-programmées, surveillance d'évènements internes, ...) a été un objectif majeur depuis la naissance des premiers véhicules spatiaux. Dans le cadre des travaux menés dans cette thèse, il s'agit de contribuer à l'amélioration de l'autonomie opérationnelle de l'engin, où le système spatial gère ses activités en fonction des spécifications de haut niveau, en termes de surveillance de son intégrité physique, planification de manoeuvres, ...

L'étude de stratégies de guidage et de pilotage avancées (G&C) pour les véhicules réutilisables de nouvelle génération a conduit à un grand nombre de percées scientifiques et technologiques dans ce domaine. Le développement de nouvelles solutions G&C fait le plus souvent appel aux techniques robustes de l'Automatique moderne, afin de tenir compte des contraintes opérationnelles, de plus en plus exigeantes, auxquels seront soumis les lanceurs de nouvelle génération.

La première partie de ce mémoire a été consacrée à la présentation des outils théoriques qui seront utilisés dans la seconde partie pour concevoir et mettre en place des stratégies innovantes de guidage autonome pour les véhicules de rentrée atmosphérique. Ces outils, introduits par l'école Française au début des années quatre-vingt-dix (voir l'article fondateur [1]), offrent un cadre méthodologique général pour l'analyse d'une classe de systèmes non linéaires, communément appelés systèmes (différentiellement) plats. Il s'agit d'une approche originale permettant d'identifier et de caractériser les systèmes "pseudo" non linéaires, c'est à dire ceux qui sont équivalents, sous certaines hypothèses, à des systèmes linéaires par transformation et bouclage. Cette théorie a connu un essor important ces dernières années et a donné lieu à de nombreux travaux portant sur la commande [2, 3], la planification de trajectoires [4, 5] ou l'identification [6], ...

---

## Organisation de la thèse

Cette thèse est divisée en deux parties, composées chacune de trois chapitres.

La **première partie**, comprenant les chapitres 1, 2 et 3, est consacrée à la caractérisation formelle de sorties plates pour les systèmes non linéaires possédant la propriété d'être différentiellement plats.

Dans le **premier chapitre** de cette partie, après un tour d'horizon rapide sur le concept de platitude, nous illustrerons l'intérêt de ces outils sur quelques exemples académiques. Ce tour d'horizon n'est certes pas complet et le bon niveau de détails n'est pas facile à trouver ; nous inviterons souvent le lecteur intéressé à se référer aux références bibliographiques dans lesquelles il pourra trouver de plus amples détails. Nous nous efforcerons de présenter, de façon la plus synthétique possible, les principaux outils et de mettre en évidence leur intérêt pour les développements ultérieurs dans ce mémoire.

Dans le **deuxième chapitre**, nous étendrons les conditions nécessaires et suffisantes de platitude établies dans [7] à la classe des systèmes linéaires différentiels à retards. A cette fin, nous proposerons un algorithme constructif, basé sur des outils classiques de l'algèbre linéaire polynomiale, permettant de vérifier si un système linéaire à retards est  $\delta$ -plat et, le cas échéant, de calculer des sorties  $\delta$ -plates candidates.

Le **troisième chapitre** est consacré à l'application des conditions nécessaires et suffisantes de platitude à un modèle à trois degrés de liberté d'un véhicule de rentrée atmosphérique. Nous verrons alors, en utilisant un algorithme de calcul formel, qu'une sortie plate candidate est constituée de la position du centre de gravité du véhicule dans un repère local lié à la piste.

La **seconde partie** de ce mémoire est consacrée au développement d'une loi de guidage autonome à base de modèles pour les lanceurs réutilisables de nouvelle génération lors de certaines phases critiques d'une mission de rentrée atmosphérique.

Le **premier chapitre** est consacré à la présentation des principales techniques d'optimisation et de planification de trajectoires de rentrée atmosphérique existant dans la littérature.

Dans le **deuxième chapitre**, nous proposerons une méthodologie originale à base de modèles pour le développement d'une loi de guidage autonome, permettant un traitement des données à bord et possédant la capacité d'être tolérante à certains

types de défauts actionneurs. Dans un premier temps, nous nous attacherons à présenter l'architecture retenue pour la loi de guidage autonome. Dans un second temps, nous utiliserons la propriété de platitude différentielle dans le cadre de la planification de trajectoires à bord. Nous verrons notamment que l'intérêt majeur de cette technique réside dans la réduction importante du nombre de variables d'optimisation intervenant dans le problème de commande optimal, mais ceci, le plus souvent, au prix d'une perte de la convexité du problème initial. Nous proposerons alors une méthode d'annexion superellipsoïdale, basée sur l'optimisation de formes géométriques déformables par un algorithme génétique, afin d'obtenir un problème de commande optimale convexe. Nous intégrerons des contraintes liées à l'équilibrage statique du véhicule dans le problème de commande optimal afin de garantir, en présence d'une situation défaillante, l'appartenance de la trajectoire replanifiée, si elle existe, à l'enveloppe de vol réduite du véhicule. Finalement, nous décrirons la méthode utilisée pour transformer le problème de commande optimale en un problème de programmation non linéaire (NLP), lequel pourra ensuite être résolu en temps réduit grâce à un solveur adapté.

Dans le **troisième chapitre**, nous nous attacherons à appliquer étape par étape, et de manière détaillée, l'ensemble de la méthodologie de guidage autonome et de planification de trajectoires proposée. Nous focaliserons cette application sur les phases de dissipation d'énergie en zone terminale (TAEM), et d'approche et d'atterrissage automatique (A&L). Nous utiliserons à cette fin un simulateur à six degrés de liberté de la navette spatiale américaine STS-1, développé dans le cadre de cette étude. Nous commencerons par présenter la problématique associée à une mission de rentrée atmosphérique en terme de guidage et de pilotage, avant d'appliquer la méthodologie proposée dans le chapitre 2. Nous montrerons en outre comment tenir compte de perturbations aérologiques, plus précisément de champs de cisaillement de vent, dans l'algorithme de planification de trajectoires.

## Conseils de lecture

Pour la rédaction de ce mémoire de thèse, nous avons opté pour une présentation détaillée et complète, tout en gardant, bien entendu, une cohérence d'ensemble et en mettant en évidence nos contributions propres. Ce choix a été motivé par la volonté de rendre le manuscrit autonome, étant donné le large spectre des thèmes abordés. De ce fait, certains passages pourraient paraître, de prime abord, un peu fastidieux à lire, si le thème est en dehors du champ de compétences du lecteur. C'est pourquoi, nous terminons cette introduction en conseillant un guide de lecture simple.

Le lecteur intéressé uniquement par les développements théoriques liés à la platitude des systèmes peut se contenter de la première partie de ce manuscrit. De plus, le lecteur familier avec les concepts associés à la platitude différentielle peut directement aborder le second chapitre de la première partie. En revanche, le lecteur étant uniquement intéressé par l'application des techniques de platitude au développement d'une loi de guidage au-

---

tonome peut directement aborder la seconde partie de ce mémoire. En outre, le lecteur familier avec les méthodes d'optimisation non linéaires et les techniques d'optimisation de trajectoires peut commencer par le chapitre 2 de la deuxième partie. Enfin, le lecteur intéressé par la chaîne complète devra, bien entendu, lire l'ensemble des chapitres des deux parties.





## Première partie

### Caractérisation formelle de sorties plates pour les systèmes non linéaires



# Position du problème

Bien que, de par leur nature, de nombreux processus industriels présentent des non linéarités statiques et dynamiques plus ou moins fortes, les méthodes de contrôle<sup>5</sup> provenant de la théorie de la commande linéaire sont souvent appliquées avec succès sur la base de modèles linéaires approchés du comportement réel du système. Il est maintenant largement reconnu dans la communauté automatique que l'utilisation de la commande non linéaire peut améliorer de manière significative les niveaux de performance ou de robustesse des systèmes commandés<sup>6</sup>, ou même parfois résoudre des cas pathologiques. La théorie de la commande non linéaire étant construite dans un cadre mathématique moins conventionnel, un effort de recherche important se poursuit depuis une vingtaine d'années avec pour ambition de développer des outils de calcul formel génériques, efficaces et pratiques en vue d'applications industrielles. Dans cette direction, plusieurs techniques de linéarisation par retour d'état et bouclage ont été introduites dans la littérature [2, 8, 9, 3, 10]. L'objectif visé était alors d'étendre, aux systèmes non linéaires, les techniques de commandes matures et éprouvées, développées dans un contexte linéaire.

De manière générale, les systèmes non linéaires peuvent être séparés en deux classes :

- Les “vrais” systèmes non linéaires, pour lesquels des outils spécifiques d'analyse et de calcul sont nécessaires,
- Les “pseudo” systèmes non linéaires qui sont équivalents, sous certaines hypothèses, à des systèmes linéaires par transformation et bouclage.

Le concept de platitude différentielle, introduit dans [1] au milieu des années 90, a pour objectif de caractériser cette seconde classe de systèmes. Depuis, ce concept a inspiré une littérature abondante, et a été appliqué à un grand nombre de problèmes :

- Commande robuste [2, 8, 9, 3, 10],
- Commande tolérante aux défauts [11],
- Planification de trajectoires [4, 5, 12, 13],
- Diagnostic de pannes [14, 15],
- Estimation non linéaire de paramètres [6].

Conceptuellement, un système non linéaire régi par des équations différentielles ordinaires est différentiellement plat (ou simplement plat) si son comportement dynamique peut être décrit au moyen d'un ensemble de fonctions non linéaires lisses et différentiellement in-

---

<sup>5</sup>au sens large, incluant les tâches de pilotage (boucles internes) et/ou guidage (boucles externes).

<sup>6</sup>dans la suite, nous appelons système commandé un système qui n'est pas autonome.

dépendantes, appelées sorties plates. Ces sorties dépendent des états du système et d'un nombre fini de dérivées temporelles des entrées. Ainsi, toute trajectoire d'un système non linéaire dans l'espace d'état peut être déterminée uniquement à partir de la connaissance de cet ensemble de fonctions, sans avoir recours à une intégration explicite de la dynamique du système. L'intérêt majeur de la platitude est lié à la détermination d'un ensemble de sorties plates assurant qu'une inversion dynamique non linéaire (NDI) du système n'engendre pas de dynamique des zéros non observable. En effet, lorsque des sorties arbitraires sont choisies de manière à réaliser une linéarisation entrées-états, une question essentielle demeure sur l'instabilité potentielle de la dynamique rendue non observable [2].

Les conditions nécessaires et suffisantes de platitude pour les systèmes linéaires et non linéaires multidimensionnels, régis par des équations différentielles ordinaires, sont bien établies [7, 16]. La contribution principale de cette partie est le développement d'un algorithme constructif permettant calculer des sorties  $\delta$ -plates candidates pour les systèmes linéaires à retards. Cet algorithme peut être vu comme une extension, pour les systèmes linéaires à retards, des conditions nécessaires et suffisantes de platitude obtenues dans [7]. Dans le troisième chapitre, nous montrerons la propriété de platitude du modèle à trois degrés de liberté d'un véhicule de rentrée atmosphérique, et nous calculerons de manière formelle la sortie plate associée. Ceci permettra d'établir le lien avec la seconde partie de ce mémoire, dédiée au développement d'une loi de guidage autonome par platitude pour les véhicules de rentrée atmosphérique.

# Chapitre 1

## Platitude différentielle

“La musique est une mathématique sonore, la  
mathématique une musique silencieuse.”

**Édouard Herriot**

---

### Résumé :

*L’objectif de chapitre est de présenter les conditions nécessaires et suffisantes de platitude établies dans un contexte algébrique différentiel.*

*Dans un premier temps, nous nous attacherons à rappeler le concept de platitude différentielle pour les systèmes non linéaires régis par des équations différentielles ordinaires, ainsi que pour les systèmes linéaires à retards.*

*Dans un second temps, nous présenterons les principaux algorithmes de calcul formel permettant de caractériser des sorties plates candidates pour les systèmes non linéaires régis par des équations différentielles ordinaires.*

---

## 1.1 Introduction

La théorie de la platitude consiste à déterminer si un système sous-déterminé d'équations peut être paramétré, de manière injective, au moyen de fonctions arbitraires. Historiquement, les premiers travaux sur la platitude ont été réalisés dans [3], en vue de traiter des applications aéronautiques. Le développement de la théorie s'est poursuivi dans le cadre des travaux de thèse de P. Martin [17], qui ont permis d'aboutir au concept de platitude formalisé dans [1]. D'autre part, la propriété de platitude des systèmes non linéaires à retards a d'abord été établie dans les travaux de thèse de H. Mounier [18], puis dans [19] et [20]. D'autres approches ont été développées dans [21, 22, 8, 23, 24, 25].

La platitude de différentes classes de systèmes (systèmes d'équations différentielles ordinaires, à retards, de dimension infinie,...) peut être décrite en utilisant différents formalismes mathématiques, reposant sur des notions d'algèbre différentielle ou de géométrie différentielle.

## 1.2 Rappels sur le concept de platitude

Nous commençons ici par rappeler la définition usuelle d'un système plat, telle qu'introduite par Fliess *et al.* dans [1] pour les systèmes non linéaires régis par des équations différentielles ordinaires (ODE). Conceptuellement, un système non linéaire est plat s'il existe un ensemble de variables différentiellement indépendantes, dont le nombre est égal au nombre de commandes, tel que l'état et les commandes du système puissent s'exprimer exclusivement en fonction de ces sorties et d'un nombre fini de leur dérivées. De ce fait, aucune intégration d'équations différentielles n'est rendue nécessaire : les trajectoires des états et des entrées du système peuvent être immédiatement obtenues à partir des trajectoires des sorties plates. Cette propriété fondamentale des systèmes plats est à la base d'applications importantes, notamment en termes de planification [4, 5] et de poursuite de trajectoires [2, 9, 3].

Plus précisément, considérons un système non linéaire déterministe défini sur une variété différentiable  $X$ , s'écrivant sous la forme suivante :

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

où  $x$  est le vecteur d'état de dimension  $n$  et  $u$  le vecteur de commandes de dimension  $m$ . De plus,  $f(\cdot)$  et  $h(\cdot)$  représentent des applications non linéaires lisses<sup>7</sup> sur  $X$ . On suppose en outre que  $m \leq n$ , c'est à dire que le système d'équations différentielles (1.1) est sous-déterminé.

**Définition 1.1** (Platitude différentielle [1]). *Le système non linéaire (1.1) est différentiellement plat (ou, plus brièvement, plat) si et seulement s'il existe un vecteur  $z$  de dimension  $m$ , dont les éléments sont des fonctions différentiellement indépendantes, de la forme :*

$$z = \phi(x, u, \dot{u}, \dots, u^{(\alpha)}),$$

---

<sup>7</sup>c'est à dire suffisamment différentiables sur  $X$ .

tel que

$$\begin{cases} x = \psi_x(z, \dot{z}, \dots, z^{(\beta-1)}) \\ u = \psi_u(z, \dot{z}, \dots, z^{(\beta)}) \end{cases}$$

où  $\psi_x$  et  $\psi_u$  représentent des applications lisses sur la variété  $X$ ,  $z_i^{(k)}$  correspond à la dérivée d'ordre  $k$  de la  $i^{\text{ème}}$  composante de  $z$ , et  $\alpha = (\alpha_1, \dots, \alpha_m)$ ,  $\beta = (\beta_1, \dots, \beta_m)$  sont des  $m$ -uplets finis d'entiers.

Le vecteur  $z \in \mathbb{R}^m$  est appelé sortie plate<sup>8</sup> (ou sortie linéarisante).

**Exemple 1.1.** Afin d'illustrer le concept de platitude pour les systèmes implicites, considérons l'exemple d'un véhicule non holonome roulant sans glisser sur un plan [26]. Dans le

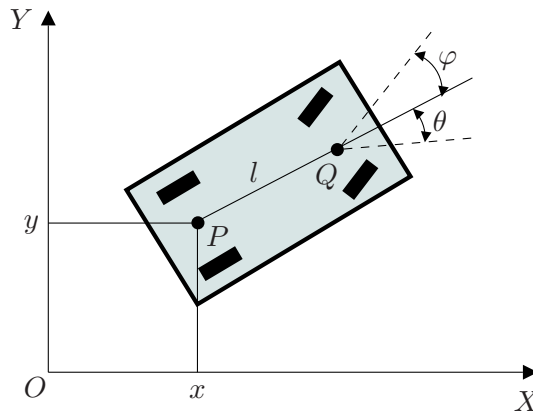


FIG. 1.1 – véhicule roulant sans glisser sur un plan.

plan  $(O, x, y)$ , notons par  $(x, y)$  les coordonnées du point  $P$ , milieu de l'essieu arrière, par  $Q$  le point situé au milieu de l'essieu avant,  $\|\overrightarrow{PQ}\| = l$ , par  $\theta$  l'angle entre l'axe du véhicule et l'axe  $Ox$ , et par  $\varphi$  l'angle de braquage des roues (voir figure 1.1). Les conditions de roulement sans glissement correspondent au fait que  $\frac{d\overrightarrow{OP}}{dt}$  reste colinéaire à  $\overrightarrow{PQ}$  et que  $\frac{d\overrightarrow{OQ}}{dt}$  reste colinéaire aux roues avant. De plus, notons  $u = \left\| \frac{d\overrightarrow{OP}}{dt} \right\| \cdot \frac{\|\overrightarrow{PQ}\|}{\|\overrightarrow{PQ}\|}$ . Les équations de la cinématique s'écrivent alors :

$$\begin{aligned} \dot{x} &= u \cos \theta \\ \dot{y} &= u \sin \theta \\ \dot{\theta} &= \frac{u}{l} \tan \varphi \end{aligned} \tag{1.2}$$

La mise sous forme implicite de ce système conduit à

$$\dot{x} \sin \theta - \dot{y} \cos \theta = 0.$$

<sup>8</sup>du fait que la sortie  $z$  ainsi que ses dérivées successives vivent dans un espace vectoriel de dimension finie, sans contraintes, donc sans courbure au sens de la géométrie différentielle.



Montrons que ce système est plat et admet la position cartésienne  $(x, y)$  du véhicule comme sortie plate. En combinant les deux premières équations de (1.2), on obtient

$$\tan \theta = \frac{\dot{y}}{\dot{x}}, \quad u^2 = \dot{x}^2 + \dot{y}^2.$$

En dérivant l'expression de  $\tan \theta$ , il vient  $\dot{\theta}(1 + \tan^2 \theta) = \frac{\ddot{y}\dot{x} - \dot{y}\ddot{x}}{\dot{x}^2}$ , d'où l'on extrait

$$\dot{\theta} = \frac{\ddot{y}\dot{x} - \dot{y}\ddot{x}}{\dot{x}^2 + \dot{y}^2}$$

et, en utilisant la troisième équation de (1.2), on peut alors exprimer la commande  $\varphi$  en fonction de la sortie plate et d'un nombre fini de ses dérivées :

$$\varphi = \arctan \left( \frac{l\dot{\theta}}{u} \right) = \arctan \left( \frac{l(\ddot{y}\dot{x} - \dot{y}\ddot{x})}{(\dot{x}^2 + \dot{y}^2)^{\frac{3}{2}}} \right).$$

Ainsi, tous les états et les commandes du système (1.2) s'expriment en fonction de  $x, \dot{x}, \ddot{x}, y, \dot{y}, \ddot{y}$ , ce qui prouve que ce dernier est plat.

**Exemple 1.2.** Considérons maintenant un système linéaire de type masses-ressorts, constitué de deux solides de masses respectives  $m_1$  et  $m_2$ , accrochés à deux ressorts de raideurs respectives  $k_1$  et  $k_2$ , pouvant se déplacer le long d'un axe horizontal (voir figure 1.2). Les abscisses respectives des centres de gravité  $G_1$  et  $G_2$  des deux solides sont notées

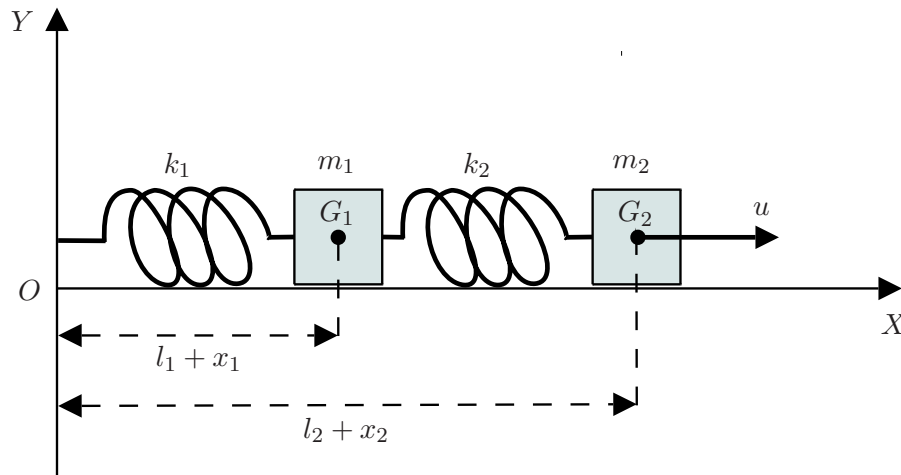


FIG. 1.2 – système masses-ressorts.

$l_1 + x_1$  et  $l_2 + x_2$ , où  $l_1$  et  $l_2$  représentent les positions, au repos, de  $G_1$  et de  $G_2$ . On suppose que les deux masses sont soumises à des frottements visqueux de coefficients respectifs  $r_1$  et  $r_2$ . Enfin, le système global est contrôlé en exerçant une force  $u$  sur  $G_1$ . Un modèle de ce dernier est donné par :

$$\begin{cases} m_1 \ddot{x}_1 + k_1 x_1 + r_1 \dot{x}_1 = k_2 (x_2 - x_1) \\ m_2 \ddot{x}_2 + k_2 (x_2 - x_1) + r_2 \dot{x}_2 = u \end{cases} \quad (1.3)$$

La forme implicite correspondante est obtenue en éliminant la dernière équation du modèle (1.3), soit

$$m_1\ddot{x}_1 + k_1x_1 + r_1\dot{x}_1 - k_2(x_2 - x_1) = 0.$$

Montrons que l'on peut exprimer  $x_2$  et  $u$  en fonction de  $x_1$ . A partir de la première équation de (1.3), on tire

$$x_2 = \frac{m_1}{k_2}\ddot{x}_1 + \frac{1}{k_2}((k_1 + k_2)x_1 + r_1\dot{x}_1). \quad (1.4)$$

En dérivant (1.4), on obtient

$$\dot{x}_2 = \frac{m_1}{k_2}\dot{x}_1^{(3)} + \frac{1}{k_2}((k_1 + k_2)\dot{x}_1 + r_1\ddot{x}_1).$$

En utilisant la seconde équation de (1.3) et les équations de  $x_2$  et  $\dot{x}_2$  précédentes, on peut extraire la commande  $u$  telle que

$$\begin{aligned} u &= \frac{m_1m_2}{k_2}x_1^{(4)} + \frac{m_2r_1 + m_1r_2}{k_2}x_1^{(3)} + \left(\frac{m_2k_1 + r_1r_2}{k_2} + m_2 + m_1\right)\ddot{x}_1 \\ &+ \left(\frac{r_2k_1}{k_2} + r_1 + r_2\right)\dot{x}_1 - k_2x_1. \end{aligned}$$

Ainsi,  $x_2$  et  $u$  s'expriment en fonction de  $x_1$  et d'un nombre fini de ses dérivées. Le système (1.3) est donc plat, avec  $x_1$  comme sortie plate.

Rappelons maintenant le concept de platitude pour les systèmes linéaires à retards. Considérons un système linéaire à retards régi par un ensemble d'équations différentielles retardées, tel que

$$\dot{x}_i = \sum_{j=1}^n a_{i,j}\left(\frac{d}{dt}, \delta\right)x_j + \sum_{j=1}^m b_{i,j}\left(\frac{d}{dt}, \delta\right)u_j, \quad i = 1, \dots, n, \quad (1.5)$$

où  $x \in \mathbb{R}^n$  est l'état,  $u \in \mathbb{R}^m$  le vecteur de commandes, et  $\delta = (\delta_1, \delta_2, \dots, \delta_r) \in \mathbb{R}^r$  un ensemble non vide d'opérateurs de retard vérifiant la loi  $\delta_i : f(t) \mapsto f(t - \tau_i)$ ,  $\tau_i \in \mathbb{R}^+$ ,  $i = 1, \dots, r$ . Les termes  $a_{i,j}(\frac{d}{dt}, \delta)$  et  $b_{i,j}(\frac{d}{dt}, \delta)$  sont définis sur l'anneau  $\mathfrak{K}[\frac{d}{dt}, \delta]$  des polynômes en  $\frac{d}{dt}$  et  $\delta$ , et à coefficients dans le champ  $\mathfrak{K}$  des fonctions méromorphes du temps, de  $\mathbb{R}^n \times \mathbb{R}^m$  dans  $\mathbb{R}$ . La loi de multiplication sur l'anneau  $\mathfrak{K}[\frac{d}{dt}, \delta]$  est définie telle que l'on a, pour tout  $a(t) \in \mathfrak{K}$ ,

$$\frac{d}{dt}a(t) = a(t)\frac{d}{dt} + \dot{a}(t), \quad \delta_i a(t) = a(t - \tau_i)\delta_i, \quad i = 1, \dots, r.$$

En notant par  $\mathcal{M}_{p,q}(\frac{d}{dt}, \delta)$  le module des matrices de taille  $p \times q$  à éléments dans l'anneau commutatif  $\mathfrak{K}[\frac{d}{dt}, \delta]$ , le système (1.5) peut être réécrit de manière plus compacte sous la forme

$$\dot{x} = Ax + Bu, \quad (1.6)$$

où  $A \in \mathcal{M}_{n,n}(\frac{d}{dt}, \delta)$  et  $B \in \mathcal{M}_{n,m}(\frac{d}{dt}, \delta)$ .

**Définition 1.2** ( $\delta$ -platitude [20]). *Le système linéaire à retards (1.5) est dit différentiellement  $\delta$ -plat (ou  $\delta$ -libre) s'il existe un polynôme non nul  $\pi = \delta_1^{s_1} \dots \delta_\rho^{s_\rho} \in \mathfrak{K}[\delta]$ ,  $s_j \in \mathbb{N}^r$ ,  $j = 1, \dots, \rho$ , et une collection  $z$  de  $m$  variables indépendantes  $\delta$ -différentiellement indépendantes<sup>9</sup>, de la forme*

$$z = P_0(\delta)x + P_1(\delta)u + P_2(\delta)\dot{u} + \dots + P_\alpha(\delta)u^{(\alpha)}, \quad (1.7)$$

où  $P_0(\delta) \in (\mathfrak{K}[\delta])^{m \times n}$ ,  $P_i(\delta) \in (\mathfrak{K}[\delta])^{m \times m}$ ,  $i = 1, \dots, \alpha$ , et  $\alpha = (\alpha_1, \dots, \alpha_m)$  est un  $m$ -uplet fini d'entiers, telle que

$$x = Q_0(\delta, \pi^{-1})z + Q_1(\delta, \pi^{-1})\dot{z} + \dots + Q_\nu(\delta, \pi^{-1})z^{(\nu)}, \quad (1.8)$$

$$u = R_0(\delta, \pi^{-1})z + R_1(\delta, \pi^{-1})\dot{z} + \dots + R_{\nu+1}(\delta, \pi^{-1})z^{(\nu+1)}, \quad (1.9)$$

où  $Q_i(\delta, \pi^{-1}) \in (\mathfrak{K}[\delta, \pi^{-1}])^{n \times m}$ ,  $i = 0, \dots, \nu$ ,  $R_j(\delta, \pi^{-1}) \in (\mathfrak{K}[\delta, \pi^{-1}])^{m \times m}$  et  $j = 0, \dots, \nu + 1$ . En outre,  $z_i^{(k)}$  désigne la dérivée temporelle d'ordre  $k$  de la  $i^{\text{ème}}$  composante de  $z$ , et  $\nu = (\nu_1, \dots, \nu_m)$  représente un  $m$ -uplet fini d'entiers.

De plus, si une telle collection  $z$  existe, elle est appelée sortie  $\delta$ -plate du système.

En d'autres termes, la définition 1.2 établit que la sortie  $\delta$ -plate  $z$  peut être obtenue comme une combinaison  $\mathfrak{K}[\frac{d}{dt}, \delta]$ -linéaire des variables du système, et que l'état et les commandes du système peuvent être calculés à partir de  $z$  en utilisant des différentiations, des retards et des avances (provenant de l'inversion du  $\pi$ -polynôme).

**Exemple 1.3.** Le système  $\dot{x}(t) = u(t-1)$  peut être naturellement considéré comme  $\delta$ -plat avec  $x$  pour sortie  $\delta$ -plate car l'entrée  $u(t) = \dot{x}(t+1)$  s'exprime en fonction de  $x$  et plus précisément de sa dérivée première avancée de 1. Par contre, si l'on interdit l'utilisation de l'opérateur d'avance, ce système ne peut plus être considéré comme  $\delta$ -plat, alors que l'utilisation de l'opérateur d'avance n'a aucune incidence sur la planification de trajectoires notamment.

**Remarque 1.1.** Pour l'exemple 1.3, le module correspondant n'est pas libre au sens de [24]. En revanche, l'équation  $u(t) = \dot{x}(t+1)$  montre qu'il est  $\delta$ -libre.

**Remarque 1.2.** La définition 1.2 constitue un cas particulier du concept plus général de  $\pi$ -liberté, où  $\pi$  peut être un polynôme arbitraire de  $\mathfrak{K}[\delta]$ . De plus, si  $\pi = \emptyset$ , on dit alors que le système (1.6) est simplement plat et, dans ce cas, que  $z$  est une sortie plate de (1.6).

Un système à retards est  $\delta$ -plat s'il est  $\mathfrak{K}[\frac{d}{dt}, \delta]$ -commandable libre. Un critère simple peut être utilisé afin de vérifier si le système (1.6) est  $\mathfrak{K}[\frac{d}{dt}, \delta]$ -commandable libre.

**Proposition 1.1** ([18]). *Soit  $P = (\frac{d}{dt}I_n - A, -B) \in \mathcal{M}_{n+m, n}(\frac{d}{dt}, \delta)$  la matrice de présentation associée au système à retards (1.6), à éléments dans l'anneau commutatif  $\mathfrak{K}[\frac{d}{dt}, \delta]$ , et supposons qu'elle soit de plein rang ligne (i.e.  $n+m$ ). Alors, (1.6) est  $\mathfrak{K}[\frac{d}{dt}, \delta]$ -commandable libre si et seulement si*

$$\forall (s, z_1, \dots, z_r) \in \overline{\mathfrak{K}}^{r+1}, \quad \text{rang}_{\overline{\mathfrak{K}}} P(s, z_1, \dots, z_r) = n + m,$$

où le corps  $\overline{\mathfrak{K}}$  désigne la clôture algébrique de  $\mathfrak{K}$ .

---

<sup>9</sup>plus précisément, il n'existe aucune relation  $\delta$ -différentielle entre les composantes de  $z$ , i.e.,  $S_0(\delta)z + S_1(\delta)\dot{z} + \dots + S_\mu(\delta)z^{(\mu)} = 0$  implique que  $S_i(\delta) = 0$ , où  $S_i(\delta) \in (\mathfrak{K}[\delta])^{(n+m) \times m}$ ,  $i = 0, \dots, \mu$ .

Le critère de  $\mathfrak{K}[\frac{d}{dt}, \delta]$ -commandabilité libre défini par la proposition 1.1 est équivalent au fait que dans  $\overline{\mathfrak{K}}^{r+1}$ , il n'y a pas de zéros communs des mineurs de  $P$  d'ordre  $n + m$ .

**Exemple 1.4.** Considérons à présent un système linéaire “hybride” constitué d'une corde au milieu de laquelle est attachée une masse  $M$  [27] (cf figure 1.3). Ce système peut être modélisé par une équation d'onde mono-dimensionnelle pour chaque intervalle entre le bord et la masse, couplée à une équation différentielle ordinaire linéaire décrivant le mouvement de la masse. Les équations d'ondes s'écrivent

$$\rho_1 \partial_t^2 p = \tau_1 \partial_x^2 p, \quad x \in [-L_1, 0], \quad t \in \mathbb{R}^+, \quad (1.10)$$

$$\rho_2 \partial_t^2 q = \tau_2 \partial_x^2 q, \quad x \in [0, L_2], \quad t \in \mathbb{R}^+, \quad (1.11)$$

où  $p(x, t)$  et  $q(x, t)$  représentent la déformation à l'abscisse  $x$  et au temps  $t$  pour chaque segment  $[-L_1, 0]$  et  $[0, L_2]$  de la corde, supposés être homogènes. Les paramètres  $\rho_1$  et  $\rho_2$  représentent la densité de chaque segment, et  $\tau_1$  et  $\tau_2$  leurs tensions respectives. En outre, l'équation différentielle régissant la position  $z(t)$  de la masse  $M$ , attachée à la corde pour  $x = 0$ , est définie par

$$z(t) = p(0, t) = q(0, t), \quad t \in \mathbb{R}^+, \quad (1.12)$$

$$M \partial_t^2 z(t) + \tau_1 \partial_x p(0, t) - \tau_2 \partial_x q(0, t) = 0, \quad t \in \mathbb{R}^+. \quad (1.13)$$

Les commandes de position  $u(t)$  et  $v(t)$  sont appliquées au bords (voir figure 1.3) et conduisent à des conditions de bord de type Dirichlet :

$$p(-L_1, t) = u(t), \quad t \in \mathbb{R}^+, \quad (1.14)$$

$$q(L_2, t) = v(t), \quad t \in \mathbb{R}^+. \quad (1.15)$$

Les conditions initiales sont définies par

$$p(x, 0) = p^0(x), \quad \partial_t p(x, 0) = p^1(x), \quad x \in [-L_1, 0], \quad (1.16)$$

$$q(x, 0) = q^0(x), \quad \partial_t q(x, 0) = q^1(x), \quad x \in [0, L_2], \quad (1.17)$$

$$z(0) = z^0 = p^0(0) = q^0(0), \quad (1.18)$$

$$\partial_t z(0) = z^1 = p^1(0) = q^1(0). \quad (1.19)$$

En utilisant le calcul opérationnel de Mikusiński (voir par exemple [28]), on peut alors transformer le système de dimension infinie (1.10)-(1.19) en un système différentiel à retards. La solution générale des équations (1.10)-(1.19) est donnée dans l'anneau de convolution  $\mathcal{D}'(\mathbb{R})$  des distributions par

$$p(x, t) = (\delta_{\mu_1 x} * \phi_1)(t) + (\delta_{-\mu_1 x} * \psi_1)(t), \quad (1.20)$$

$$q(x, t) = (\delta_{\mu_2 x} * \phi_2)(t) + (\delta_{-\mu_2 x} * \psi_2)(t), \quad (1.21)$$

avec  $\phi_1, \phi_2, \psi_1, \psi_2 \in \mathcal{D}'(\mathbb{R})$ , et où  $\delta_h$  désigne la distribution de Dirac concentrée en  $h$ . De plus,

$$\mu_1 = \sqrt{\frac{\rho_1}{\tau_1}}, \quad \mu_2 = \sqrt{\frac{\rho_2}{\tau_2}}.$$

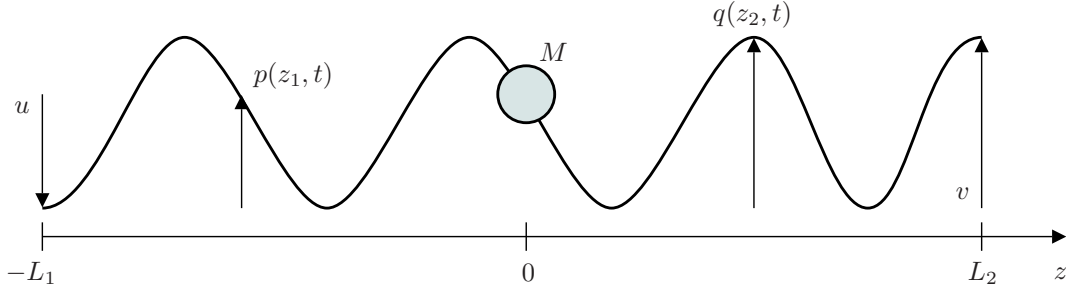


FIG. 1.3 – corde vibrante avec une masse.

Les conditions de bord (1.12), (1.14) et (1.15) s'écrivent alors

$$\phi_1(t) + \psi_1(t) - \phi_2(t) - \psi_2(t) = 0, \quad (1.22)$$

$$M(\delta'' * (\phi_1 + \psi_1)(t)) + \tau_1 \mu_1 (\delta' * (\phi_1 - \psi_1)(t)) - \tau_2 \mu_2 (\delta' * (\phi_2 - \psi_2)(t)) = 0 \quad (1.23)$$

$$(\delta_{-\mu_1 L_1} * \phi_1)(t) + (\delta_{\mu_1 L_1} * \psi_1)(t) = u(t), \quad (1.24)$$

$$(\delta_{\mu_2 L_2} * \phi_2)(t) + (\delta_{-\mu_2 L_2} * \psi_2)(t) = v(t), \quad (1.25)$$

où  $\delta'$  et  $\delta''$  représentent les dérivées de la distribution de Dirac concentrées en 0. Comme  $\bar{\phi}_1(t) = \phi_1(t) + \alpha(t)$  et  $\bar{\psi}_1(t) = \psi_1(t) - \alpha(t)$  engendrent la même solution pour tout  $\alpha(t)$ , on peut alors choisir

$$M(\delta' * (\phi_1 + \psi_1)(t)) + \tau_1 \mu_1 (\phi_1 - \psi_1)(t) - \tau_2 \mu_2 (\phi_2 - \psi_2)(t) = 0. \quad (1.26)$$

Maintenant, en posant

$$\eta_1 = \frac{\tau_1 \mu_1}{M}, \quad \eta_2 = \frac{\tau_2 \mu_2}{M},$$

et les opérateurs de retard  $\sigma_1$  et  $\sigma_2$  de durées respectives  $\mu_1 L_1$  et  $\mu_2 L_2$ , les équations précédentes deviennent

$$\begin{cases} \psi_1(t) + \phi_1(t) - \psi_2(t) - \phi_2(t) = 0, \\ \dot{\psi}_1(t) + \dot{\phi}_1(t) + \eta_1(\phi_1(t) - \psi_1(t)) - \eta_2(\phi_2(t) - \psi_2(t)) = 0, \\ \phi_1(t - \sigma_1) + \psi_1(t + \sigma_1) = u(t), \\ \phi_2(t + \sigma_2) + \psi_2(t - \sigma_2) = v(t). \end{cases} \quad (1.27)$$

En choisissant la sortie  $\delta$ -plate candidate  $z = (\psi_2, \phi_2)$ , l'état du système (1.27) peut être réécrit en fonction de cette sortie et des opérateurs de retard et d'avance, tel que

$$\begin{cases} \psi_1(t) = \frac{1}{2\eta_1}(-\dot{z}_1(t) - \dot{z}_2(t) + (\eta_1 - \eta_2)z_1(t) + (\eta_1 + \eta_2)z_2(t)), \\ \phi_1(t) = \frac{1}{2\eta_1}(\dot{z}_1(t) + \dot{z}_2(t) + (\eta_1 + \eta_2)z_1(t) + (\eta_1 - \eta_2)z_2(t)), \\ \psi_2(t) = z_1(t), \\ \phi_2(t) = z_2(t). \end{cases}$$

De la même manière, les commandes du système sont alors définies par

$$\begin{aligned} u(t) &= \frac{1}{2\eta_1}[\dot{z}_1(t - \mu_1 L_1) - \dot{z}_1(t + \mu_1 L_1) + \dot{z}_2(t - \mu_1 L_1) - \dot{z}_2(t + \mu_1 L_1) \\ &\quad + (\eta_1 + \eta_2)(z_1(t - \mu_1 L_1) + z_2(t + \mu_1 L_1)) + (\eta_1 - \eta_2)(z_1(t + \mu_1 L_1) + z_2(t - \mu_1 L_1))], \\ v(t) &= z_1(t - \mu_2 L_2) + z_2(t + \mu_2 L_2). \end{aligned}$$

## 1.3 Conditions nécessaires et suffisantes de platitude

Dans cette thèse, nous allons appliquer les techniques de calcul de sorties plates introduites dans [16, 7]. Une autre méthode, basée sur la théorie des modules, peut être trouvée dans [29]. Nous rappelons dans cette section les conditions nécessaires et suffisantes de platitude établies par J. Lévine [16] pour les systèmes non linéaires décrits par des équations différentielles ordinaires.

Soit  $X$  la variété différentiable de dimension  $n$ ,  $T_x X$  son espace tangent en un point arbitraire  $x \in X$  et  $TX = \bigcup_{x \in X} T_x X$  son fibré tangent. Considérons un système non linéaire sous forme implicite défini par

$$F(x, \dot{x}) = 0, \quad (1.28)$$

avec  $\text{rang} \frac{\partial F}{\partial \dot{x}} = n - m$  dans un voisinage donné de  $TX$ . Soit  $\mathfrak{X} \triangleq X \times \mathbb{R}_\infty^n = X \times \mathbb{R}^n \times \mathbb{R}^n \times \dots$  la variété de jets d'ordre infini, définie par le produit de  $X$  avec un nombre infini mais dénombrable de répliques de  $\mathbb{R}^n$ , et  $\bar{x} = (x_1, \dots, x_n, \dot{x}_1, \dots, \dot{x}_n, \dots, x_1^{(k)}, \dots, x_n^{(k)}, \dots)$  ses coordonnées généralisées. Le formalisme des jets d'ordre infini permet de décrire le système implicite (1.28) sous la forme suivante [16].

**Définition 1.3** (Système implicite [16]). *Un système implicite est la donnée d'un triplet  $(\mathfrak{X}, \tau_{\mathfrak{X}}, F)$  avec  $\mathfrak{X} = X \times \mathbb{R}_\infty^n$ ,  $\tau_{\mathfrak{X}} = \sum_{i \geq 0} \sum_{j=1}^n x_j^{(i+1)} \frac{\partial}{\partial x_j^{(i)}}$  le champ de Cartan trivial associé à  $\mathfrak{X}$ , et  $F \in C^\infty(TX; \mathbb{R}^{n-m})$  vérifiant  $\text{rang} \frac{\partial F}{\partial \dot{x}} = n - m$  dans un voisinage approprié de  $TX$ .*

Considérons maintenant deux systèmes implicites commandés  $(\mathfrak{X}, \tau_{\mathfrak{X}}, F)$ , avec  $\mathfrak{X} = X \times \mathbb{R}_\infty^n$ ,  $\dim X = n$ ,  $\tau_{\mathfrak{X}}$  le champ de Cartan trivial associé à  $\mathfrak{X}$ ,  $\text{rang} \frac{\partial F}{\partial \dot{x}} = n - m$ , et  $(\mathfrak{Y}, \tau_{\mathfrak{Y}}, G)$ , avec  $\mathfrak{Y} = Y \times \mathbb{R}_\infty^p$ ,  $\dim Y = p$ ,  $\tau_{\mathfrak{Y}}$  le champ de Cartan trivial associé et  $\text{rang} \frac{\partial G}{\partial \dot{y}} = p - q$ . De plus, considérons les ensembles  $\mathfrak{X}_0 = \{\bar{x} \in \mathfrak{X} \mid L_{\tau_{\mathfrak{X}}}^k F(\bar{x}) = 0, \forall k \geq 0\}$  et  $\mathfrak{Y}_0 = \{\bar{y} \in \mathfrak{Y} \mid L_{\tau_{\mathfrak{Y}}}^k G(\bar{y}) = 0, \forall k \geq 0\}$ .

**Définition 1.4** (Équivalence de Lie-Bäcklund [16]). *On dit que les deux systèmes implicites  $(\mathfrak{X}, \tau_{\mathfrak{X}}, F)$  et  $(\mathfrak{Y}, \tau_{\mathfrak{Y}}, G)$  sont Lie-Bäcklund équivalents (ou L-B équivalents) en  $(\bar{x}_0, \bar{y}_0) \in \mathfrak{X}_0 \times \mathfrak{Y}_0$  si et seulement si :*

1. *il existe des voisinages  $\mathcal{X}_0$  et  $\mathcal{Y}_0$  de  $\bar{x}_0 \in \mathfrak{X}_0$  et  $\bar{y}_0 \in \mathfrak{Y}_0$  respectivement, et une application  $\Phi = (\varphi_0, \varphi_1, \dots) \in C^\infty(\mathcal{Y}_0; \mathcal{X}_0)$  telle que  $\Phi(\bar{y}_0) = \bar{x}_0$  et  $\Phi_* \tau_{\mathfrak{Y}} = \tau_{\mathfrak{X}}$ ,*
2. *il existe une application  $\Psi = (\psi_0, \psi_1, \dots) \in C^\infty(\mathcal{X}_0; \mathcal{Y}_0)$  telle que  $\Psi(\bar{x}_0) = \bar{y}_0$  et  $\Psi_* \tau_{\mathfrak{X}} = \tau_{\mathfrak{Y}}$ .*

*Les applications  $\Phi$  et  $\Psi$  sont appelées isomorphismes de Lie-Bäcklund mutuellement inverses en  $(\bar{x}_0, \bar{y}_0)$ .*

*En outre, les deux systèmes  $(\mathfrak{X}, \tau_{\mathfrak{X}}, F)$  et  $(\mathfrak{Y}, \tau_{\mathfrak{Y}}, G)$  sont dit localement L-B équivalents s'ils sont L-B équivalents en chaque paire  $(\bar{x}, \Psi(\bar{x})) = (\Phi(\bar{y}), \bar{y})$  d'un ouvert dense  $\mathcal{Z}$  de  $\mathfrak{X}_0 \times \mathfrak{Y}_0$ , où  $\Phi$  et  $\Psi$  sont des isomorphismes de Lie-Bäcklund mutuellement inverses sur  $\mathcal{Z}$ .*

La platitude du système implicite  $(\mathfrak{X}, \tau_{\mathfrak{X}}, F)$  correspond alors au fait qu'il soit Lie-Bäcklund équivalent au système trivial  $(\mathbb{R}_{\infty}^m, \tau_m, 0)$ .

**Définition 1.5** (Platitude, formulation externe [16]). *Le système implicite  $(\mathfrak{X}, \tau_{\mathfrak{X}}, F)$  est plat en  $(\bar{x}_0, \bar{y}_0) \in \mathfrak{X}_0 \times \mathbb{R}_{\infty}^m$  si et seulement s'il est L-B équivalent, en  $(\bar{x}_0, \bar{y}_0) \in \mathfrak{X}_0 \times \mathbb{R}_{\infty}^m$ , au système implicite trivial  $(\mathbb{R}_{\infty}^m, \tau_m, 0)$ . Dans ce cas, les isomorphismes de Lie-Bäcklund  $\Phi$  et  $\Psi$  sont appelés trivialisations inverses (ou uniformisations, en référence au 22<sup>ème</sup> problème de Hilbert).*

L'extension de la définition 1.5 à la platitude locale est donnée ci-dessous.

**Théorème 1.1** (Platitude locale [16]). *Le système implicite  $(\mathfrak{X}, \tau_{\mathfrak{X}}, F)$  est plat en  $(\bar{x}_0, \bar{y}_0)$  avec  $\bar{x}_0 \in \mathfrak{X}_0$  et  $\bar{y}_0 \in \mathbb{R}_{\infty}^m$  si et seulement s'il existe une application locale inversible lisse  $\Phi$  de  $\mathbb{R}_{\infty}^m$  dans  $\mathfrak{X}_0$ , possédant une inverse lisse, satisfaisant  $\Phi(\bar{y}_0) = \bar{x}_0$ , et telle que*

$$\Phi^* dF = 0. \quad (1.29)$$

En définissant les matrices polynomiales

$$P(F) = \frac{\partial F}{\partial x} + \frac{\partial F}{\partial \dot{x}} \frac{d}{dt}, \quad P(\varphi_0) = \sum_{j \geq 0} \frac{\partial \varphi_0}{\partial y^{(j)}} \frac{d^j}{dt^j}, \quad (1.30)$$

on obtient la relation

$$\Phi^* dF = P(F)P(\varphi_0)dy = 0. \quad (1.31)$$

En utilisant des outils classiques provenant de l'algèbre linéaire polynomiale, permettant de transformer une matrice en sa forme normale de Smith (voir annexe A), le système variationnel  $P(F)$  peut être décomposé sous la forme

$$VP(F)U = (I_{n-m}, 0_{n-m,m}). \quad (1.32)$$

Notons par  $\mathfrak{K}$  le corps des fonctions méromorphes de  $\mathfrak{X}$  dans  $\mathbb{R}$ , et par  $\mathfrak{K}[\frac{d}{dt}]$  l'anneau idéal principal des polynômes en  $\frac{d}{dt}$  et à coefficients dans  $\mathfrak{K}$ . De plus,  $\mathcal{M}_{p,q}(\frac{d}{dt})$  désigne le module des matrices de dimension  $p \times q$  sur  $\mathfrak{K}[\frac{d}{dt}]$ , et  $UU_p$  l'ensemble des matrices *unimodulaires*, c'est à dire des matrices de  $\mathcal{M}_{p,p}(\frac{d}{dt})$  dont l'inverse appartient encore à  $\mathcal{M}_{p,p}(\frac{d}{dt})$ . En utilisant ces notations, l'ensemble des matrices  $P(\varphi_0)\mathcal{M}_{n,m}(\frac{d}{dt})$  vérifiant l'équation (1.31) peut être complètement caractérisé, en commençant par résoudre l'équation matricielle

$$P(F)\Theta dy = 0. \quad (1.33)$$

**Lemme 1.1** ([16]). *L'ensemble des matrices hyper-régulières  $\Theta \in \mathcal{M}_{n,m}(\frac{d}{dt})$  satisfaisant (1.33) est non vide et est donné par*

$$\Theta = U \begin{pmatrix} 0_{n-m,m} \\ I_m \end{pmatrix} W, \quad (1.34)$$

où  $U \in \text{D-Smith}(P(F))$  et  $W \in \mathcal{U}_m(\frac{d}{dt})$  est une matrice unimodulaire arbitraire.

Posons pour la suite

$$\hat{U} = U \begin{pmatrix} 0_{n-m,m} \\ I_m \end{pmatrix}. \quad (1.35)$$

**Lemme 1.2** ([16]). *Pour toute matrice  $Q \in \mathbf{G}\text{-Smith}(\hat{U})$ , il existe une matrice  $Z \in \mathcal{U}_m(\frac{d}{dt})$  telle que*

$$Q\Theta = \begin{pmatrix} I_m \\ 0_{n-m,m} \end{pmatrix} Z. \quad (1.36)$$

*De plus, pour tout  $Q \in \mathbf{G}\text{-Smith}(\hat{U})$ , la sous-matrice  $\hat{Q} = (0_{n-m,m}, I_{n-m})Q$  est équivalente à  $P(F)$ , i.e., il existe une matrice unimodulaire  $L \in \mathcal{U}_{n-m}(\frac{d}{dt})$  vérifiant  $P(F) = L\hat{Q}$ .*

Une sortie plate du système variationnel  $P(F)$  est alors donnée par

$$\omega(\bar{x}) = \begin{pmatrix} \omega_1(\bar{x}) \\ \vdots \\ \omega_m(\bar{x}) \end{pmatrix} = (I_m, 0_{m,n-m}) Q(\bar{x}) dx|_{x_0} = \tilde{Q} dx|_{x_0}. \quad (1.37)$$

Si  $d\omega = 0$ , alors une sortie plate du système non linéaire implicite (1.28) peut être obtenue en intégrant l'équation  $dy = \omega$ . Dans le cas contraire, il est nécessaire d'obtenir, lorsqu'elle existe, une base intégrable, c'est à dire un *facteur intégrant*  $M \in \mathcal{U}_m(\frac{d}{dt})$  vérifiant  $d(M\omega) = 0$ .

**Définition 1.6** (Forte fermeture [16]). *Le  $\mathfrak{K}[\frac{d}{dt}]$ -idéal  $\Omega$ , finiment généré par les 1-formes  $\omega_1, \dots, \omega_m$ , est dit fortement fermé si et seulement s'il existe une matrice  $M \in \mathcal{U}_m(\frac{d}{dt})$  telle que  $d(M\omega) = 0$ .*

Le calcul de la matrice  $M$  peut être réalisé comme suit.

**Théorème 1.2** ([16]). *Le  $\mathfrak{K}[\frac{d}{dt}]$ -idéal  $\Omega$ , finiment généré par les 1-formes  $\omega_1, \dots, \omega_m$  définies par (1.37), est fortement fermé dans  $\mathfrak{X}_0$  (où, de manière équivalente, le système  $(\mathfrak{X}, \tau_{\mathfrak{X}}, F)$  est plat) si et seulement s'il existe un opérateur  $\mu \in \mathcal{L}_1((\Lambda(\mathfrak{X}))^m)$ , et une matrice  $M \in \mathcal{U}_m(\frac{d}{dt})$  tels que*

$$d\omega = \mu\omega, \quad \mathfrak{d}(\mu) = \mu^2, \quad \mathfrak{d}(M) = -M\mu, \quad (1.38)$$

où  $\omega$  désigne le vecteur de 1-formes  $(\omega_1, \dots, \omega_m)^T$ ,  $\mathcal{L}_1((\Lambda(\mathfrak{X}))^m)$  est l'espace des opérateurs linéaires qui envoient des  $p$ -formes vectorielles de dimension  $m$  sur  $\mathfrak{X}$  en  $(p+1)$ -formes vectorielles de dimension  $m$  sur  $\mathfrak{X}$ , et  $\mathfrak{d}$  représente l'extension de la différentiation extérieure  $d$  pour les formes dont les coefficients sont à valeur dans  $\mathfrak{K}[\frac{d}{dt}]$ .

En outre, si la relation (1.38) est satisfaite, une sortie plate  $y$  est obtenue en intégrant le système d'équations  $dy = M\omega$ .

On déduit des développements précédents l'algorithme formel 1, permettant de vérifier si un système est plat et, le cas échéant, de calculer une sortie plate.



**Algorithme 1** : Calcul formel d'une sortie plate.

**Entrées** : Un système non linéaire sous forme implicite  $F(x, \dot{x}) = 0$ .

**Sorties** : Si elle existe, une sortie plate  $y$  du système non linéaire implicite  $F(x, \dot{x}) = 0$ .

**Algorithme** :

- 1 Calculer le système variationnel  $P(F)$  associé au système implicite  $F(x, \dot{x}) = 0$  tel que

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

- 2 Calculer ensuite une décomposition de Smith de  $P(F)$ , puis de  $\hat{U}$ , selon les lemmes 1.1 et 1.2. Si  $P(F)$  n'est pas hyper-régulière, alors le système n'est pas plat. Dans le cas contraire, on calcule le vecteur de 1-formes  $\omega$  défini par (1.37).
- 3 Calculer l'opérateur  $\mu$  tel que  $d\omega = \mu\omega$  par identification terme à terme.
- 4 Sinon, parmi les opérateurs  $\mu$  candidats, seuls ceux vérifiant  $\mathfrak{d}(\mu) = \mu^2$  sont conservés.
- 5 Déterminer ensuite, par identification terme à terme, une matrice  $M$  telle que  $\mathfrak{d}(M) = -M\mu$ .
- 6 Conserver uniquement les matrices  $M$  candidates étant unimodulaires. Si une telle matrice  $M$  n'existe pas, alors le système n'est pas plat. Dans le cas contraire, une sortie plate est obtenue par intégration du système d'équations  $dy = M\omega$ .

**Exemple 1.5** ([30]). Reprenons l'exemple du véhicule non holonome en dimension 2 défini dans la section 1.2. Puisque  $n = 3$  et  $m = 2$ , on a donc  $n - m = 1$  et le modèle (1.2) est équivalent à l'équation implicite scalaire obtenue en éliminant les entrées de commande  $u$  et  $\varphi$  :

$$F(x, y, \theta, \dot{x}, \dot{y}, \dot{\theta}) = \dot{x} \sin \theta - \dot{y} \cos \theta = 0. \quad (1.39)$$

Appliquons l'algorithme 1 étape par étape.

**Étape 1** : Le système variationnel  $P(F)$  associé à (1.39) est défini par

$$\begin{aligned} P(F) &= \left( \frac{\partial F}{\partial x} + \frac{\partial F}{\partial \dot{x}} \frac{d}{dt}, \quad \frac{\partial F}{\partial y} + \frac{\partial F}{\partial \dot{y}} \frac{d}{dt}, \quad \frac{\partial F}{\partial \theta} + \frac{\partial F}{\partial \dot{\theta}} \frac{d}{dt} \right) \\ &= \left( \sin \theta \frac{d}{dt}, \quad -\cos \theta \frac{d}{dt}, \quad \dot{x} \cos \theta + \dot{y} \sin \theta \right). \end{aligned} \quad (1.40)$$

**Étape 2** : Commençons par déterminer, en utilisant les lemmes 1.1 et 1.2, une sortie plate du système variationnel  $P(F)$ . Posons  $E = \dot{x} \cos \theta + \dot{y} \sin \theta$ , et appliquons la décomposition de Smith (voir annexe A, algorithme 13). On obtient alors une matrice de transformation unimodulaire  $U \in \text{D-Smith}(P(F))$  telle que

$$U = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ \frac{1}{E} & \frac{\cos \theta}{E} \frac{d}{dt} & -\frac{\sin \theta}{E} \frac{d}{dt} \end{pmatrix}.$$

Donc

$$\hat{U} = U \begin{pmatrix} 0_{1,2} \\ I_2 \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ 1 & 0 \\ \frac{\cos \theta}{E} \frac{d}{dt} & -\frac{\sin \theta}{E} \frac{d}{dt} \end{pmatrix},$$

où  $I_2$  représente la matrice identité de  $\mathbb{R}^2$ . De même, le calcul de  $Q \in \mathbf{G}\text{-Smith}(\hat{U})$  conduit à

$$Q = \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ \frac{\sin \theta}{E} \frac{d}{dt} & -\frac{\cos \theta}{E} \frac{d}{dt} & 1 \end{pmatrix}.$$

Multiplions la matrice  $Q$  par le vecteur  $(dx \quad dy \quad d\theta)^T$ . La dernière ligne s'écrit alors

$$\frac{1}{E} (\sin \theta dx - \cos \theta dy + (\dot{x} \cos \theta + \dot{y} \sin \theta) d\theta) = \frac{1}{E} d(x \sin \theta - y \cos \theta)$$

qui, avec (1.39), est identiquement nulle sur  $\mathfrak{X}_0$ .

La partie restante du système, i.e.,

$$\begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \end{pmatrix} \begin{pmatrix} dx \\ dy \\ d\theta \end{pmatrix} = \begin{pmatrix} \omega_1 \\ \omega_2 \end{pmatrix}$$

est trivialement fortement fermée avec  $M = I_2$ , ce qui donne finalement la sortie plate  $y_p = (y, x)$ . On retrouve ainsi, à une permutation près, la sortie plate obtenue dans la section 1.2.

**Étape 2 :** (*variante*) Afin d'illustrer de manière plus pertinente l'algorithme de forte fermeture du  $\mathfrak{K}[\frac{d}{dt}]$ -idéal  $\Omega$ , nous considérons une décomposition de  $P(F)$  alternative<sup>10</sup>, telle que le vecteur de 1-formes  $\omega$  correspondant ne soit pas directement intégrable. Le but est de montrer comment les équations de structure généralisées du repère mobile de Cartan (1.38) peuvent être utilisées afin d'obtenir une nouvelle base  $M\omega$  intégrable. Il suffit pour cela de reprendre la décomposition de Smith du système variationnel  $P(F)$ , en multipliant ce dernier à droite par

$$\begin{pmatrix} \cos \theta & 0 & 0 \\ \sin \theta & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix},$$

et en utilisant la formule  $\sin \theta \frac{d}{dt}(\cos \theta) - \cos \theta \frac{d}{dt}(\sin \theta) = -\dot{\theta}$ . La décomposition de Smith de  $P(F)$  à droite est alors donnée par

$$U = \begin{pmatrix} \cos \theta & -\frac{1}{\dot{\theta}} \cos^2 \theta \frac{d}{dt} & \frac{1}{\dot{\theta}} (\dot{x} \cos \theta + \dot{y} \sin \theta) \cos \theta \\ \sin \theta & 1 - \frac{1}{\dot{\theta}} \sin \theta \cos \theta \frac{d}{dt} & \frac{1}{\dot{\theta}} (\dot{x} \cos \theta + \dot{y} \sin \theta) \sin \theta \\ 0 & 0 & 1 \end{pmatrix},$$

---

<sup>10</sup>toutes les décompositions étant équivalentes dans le sens où une décomposition peut être déduite d'une autre par multiplication par une matrice unimodulaire.

et la matrice  $\hat{U}$  est définie par

$$\hat{U} = \begin{pmatrix} -\frac{1}{\theta} \cos^2 \theta \frac{d}{dt} & \frac{1}{\theta} (\dot{x} \cos \theta + \dot{y} \sin \theta) \cos \theta \\ 1 - \frac{1}{\theta} \sin \theta \cos \theta \frac{d}{dt} & \frac{1}{\theta} (\dot{x} \cos \theta + \dot{y} \sin \theta) \sin \theta \\ 0 & 1 \end{pmatrix}.$$

La décomposition de Smith à gauche de  $\hat{U}$  conduit alors à  $Q \in \mathbf{G}\text{-Smith}(\hat{U})$  avec

$$Q = \begin{pmatrix} -\tan \theta & 1 & 0 \\ 0 & 0 & 1 \\ -\frac{1}{\theta} \sin \theta \cos \theta \frac{d}{dt} & -\frac{1}{\theta} \cos^2 \theta \frac{d}{dt} & -\frac{1}{\theta} (\dot{x} \cos \theta + \dot{y} \sin \theta) \cos \theta \end{pmatrix}.$$

Le vecteur de 1-formes  $\omega$  est alors donné par

$$\omega = (\omega_1, \omega_2)^T = \hat{Q}(dx, dy, d\theta)^T = (-\tan \theta dx + dy, d\theta)^T.$$

On a

$$d\omega = (d\omega_1, d\omega_2)^T = \left(-\frac{1}{\cos^2 \theta} d\theta \wedge dx, 0\right)^T,$$

ce qui prouve que  $\omega$  n'est pas fermée.

**Étape 3 :** On introduit l'opérateur  $\mu$  le plus simple tel que

$$\mu = \begin{pmatrix} 0 & \mu_{120} \wedge \\ 0 & 0 \end{pmatrix}.$$

**Étape 4 :** On vérifie facilement que

$$\mu^2 = \begin{pmatrix} 0 & \mu_{120} \wedge \\ 0 & 0 \end{pmatrix} \begin{pmatrix} 0 & \mu_{120} \wedge \\ 0 & 0 \end{pmatrix} = 0.$$

D'autre part, on a

$$\mathfrak{d}(\mu) = \begin{pmatrix} 0 & d\left(\frac{1}{\cos^2 \theta} dx + \mu_{1230}(x, y, \theta) d\theta\right) \wedge \\ 0 & 0 \end{pmatrix},$$

où  $\mu_{1230}(x, y, \theta)$  est une fonction méromorphe arbitraire. La condition  $\mathfrak{d}(\mu) = \mu^2$  se traduit alors par le système d'EDP

$$\begin{cases} \frac{\partial}{\partial x} \mu_{1230}(x, y, \theta) = \frac{2 \sin \theta}{\cos^3 \theta}, \\ \frac{\partial}{\partial y} \mu_{1230}(x, y, \theta) = 0, \\ \frac{\partial}{\partial \theta} \mu_{1230}(x, y, \theta) = 0, \end{cases},$$

qui a pour solution

$$\mu_{1230}(x, y, \theta) = \frac{2 \sin \theta x}{\cos^3 \theta} + C_1(\theta).$$

Donc,

$$\mu = \begin{pmatrix} 0 & \left(\frac{1}{\cos^2 \theta} dx + \left(\frac{2 \sin \theta x}{\cos^3 \theta} + C_1(\theta)\right) d\theta\right) \wedge \\ 0 & 0 \end{pmatrix}. \quad (1.41)$$

**Étapes 5/6 :** La matrice unimodulaire  $M$  la plus simple est donnée par

$$\begin{pmatrix} 1 & m_{120}(x, y, \theta) \\ 0 & 1 \end{pmatrix}.$$

Calculons  $\mathfrak{d}(M) = -M\mu$ . On obtient

$$\begin{pmatrix} 1 & dm_{120} \\ 0 & 0 \end{pmatrix} = -\mu,$$

auquel correspond le système d'EDP

$$\begin{cases} \frac{\partial}{\partial x} m_{120}(x, y, \theta) = -\frac{1}{\cos^2 \theta}, \\ \frac{\partial}{\partial y} m_{120}(x, y, \theta) = 0, \\ \frac{\partial}{\partial \theta} m_{120}(x, y, \theta) = -\left(\frac{2 \sin \theta x}{\cos^3 \theta} + C_1(\theta)\right), \end{cases}$$

dont la solution est donnée par

$$m_{120} = -\frac{x}{\cos^2 \theta} + C_2(\theta),$$

où la fonction  $C_1(\theta)$  dans (1.41) est égale à  $C_1(\theta) = -\frac{d}{d\theta} C_2(\theta)$ . Il en résulte que

$$M = \begin{pmatrix} 1 & -\frac{x}{\cos^2 \theta} + C_2(\theta) \\ 0 & 1 \end{pmatrix}.$$

On en déduit alors que la sortie plate  $y_p$  doit vérifier

$$dy_p = M\omega = \begin{pmatrix} -\tan \theta dx + dy + \left(-\frac{x}{\cos^2 \theta} + C_2(\theta)\right) d\theta \\ d\theta \end{pmatrix}.$$

Cette 1-forme est bien fermée, et la sortie plate  $y_p$  est donnée par

$$y_p = (y - x \tan \theta + C_2(\theta), \theta)^T.$$

Bien sûr, la méthode décrite précédemment pour les systèmes non linéaires est applicable aux systèmes linéaires régis par des équations différentielles ordinaires [7]. Illustrons l'algorithme 1 dans le cas d'un système linéaire.

**Exemple 1.6.** Considérons à nouveau le système masses-ressort (1.3) défini dans la section 1.2. En définissant l'anneau idéal principal  $\mathfrak{R}[\frac{d}{dt}]$  des polynômes en  $\frac{d}{dt}$  et à coefficients dans le corps des fonctions méromorphes  $\mathfrak{K}$ , alors (1.3) peut se réécrire sous la forme  $A(\frac{d}{dt}) = B(\frac{d}{dt})u$ , avec

$$A = \begin{pmatrix} m_1 \frac{d^2}{dt^2} + r_1 \frac{d}{dt} + k_1 + k_2 & -k_2 \\ -k_2 & m_2 \frac{d^2}{dt^2} + r_2 \frac{d}{dt} + k_2 \end{pmatrix} \in \mathcal{M}_{2,2}\left(\frac{d}{dt}\right),$$

et

$$B = \begin{pmatrix} 0 \\ 1 \end{pmatrix} \in \mathcal{M}_{2,1}\left(\frac{d}{dt}\right).$$

Afin d'éliminer la commande  $u$ , on détermine une matrice  $M$  orthogonale à  $B$ , telle que  $M = (1, 0)$ . Le système implicite équivalent à (1.3) est donc donné par

$$MA \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \left( m_1 \frac{d^2}{dt^2} + r_1 \frac{d}{dt} + k_1 + k_2 \quad -k_2 \right) \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = 0 .$$

On cherche alors une matrice polynomiale  $\Theta \in \mathcal{M}_{n,m}(\frac{d}{dt}) = (\Theta_1 \quad \Theta_2)^T$ , solution de

$$MA\Theta = \left( m_1 \frac{d^2}{dt^2} + r_1 \frac{d}{dt} + k_1 + k_2 \quad -k_2 \right) \begin{pmatrix} \Theta_1 \\ \Theta_2 \end{pmatrix} = 0 ,$$

soit

$$m_1 \frac{d^2}{dt^2} \Theta_1 + r_1 \frac{d}{dt} \Theta_1 + (k_1 + k_2) \Theta_1 = k_2 \Theta_2 .$$

Comme les polynômes  $m_1 \frac{d^2}{dt^2} + r_1 \frac{d}{dt} + k_1 + k_2$  et  $-k_2$  sont premiers entre eux, il vient

$$\Theta_1 = 1, \quad \Theta_2 = \frac{1}{k_2} \left( m_1 \frac{d^2}{dt^2} + r_1 \frac{d}{dt} + k_1 + k_2 \right)$$

et, grâce au terme constant et égal à 1 de  $\Theta_1$ , le vecteur  $\Theta$  est clairement de rang 1 et donc surjectif.

Les états  $x$  et  $x_B$  s'expriment donc en fonction des sorties plates sous la forme

$$\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \Theta y = \begin{pmatrix} y \\ \frac{m_1}{k_2} \frac{d^2}{dt^2} y + \frac{r_1}{k_2} \frac{d}{dt} y + \frac{k_1+k_2}{k_2} y \end{pmatrix}, \quad (1.42)$$

et la force  $u$  est définie par

$$\begin{aligned} u &= \frac{m_1 m_2}{k_2} y^{(4)} + \frac{m_2 r_1 + m_1 r_2}{k_2} y^{(3)} + \left( \frac{m_2 k_1 + r_1 r_2}{k_2} + m_2 + m_1 \right) \ddot{y} \\ &+ \left( \frac{r_2 k_1}{k_2} + r_1 + r_2 \right) \dot{y} - k_2 y . \end{aligned}$$

Finalement, en inversant l'équation (1.42), on obtient trivialement la sortie plate  $y = x_1$ .

## 1.4 Conclusion

Dans ce premier chapitre, nous avons rappelé le concept de platitude pour les systèmes non linéaires régis par des équations différentielles ordinaires, ainsi que pour les systèmes linéaires à retards. Nous avons également présenté brièvement les conditions nécessaires et suffisantes de platitude pour les systèmes d'équations différentielles ordinaires.

Dans le chapitre suivant, nous abordons quelques considérations pratiques concernant les algorithmes de calcul des sorties plates. Notamment, nous proposons une extension, pour la classe des systèmes linéaires à retards, des conditions nécessaires et suffisantes de platitude établies dans [7]. Finalement, nous montrerons la propriété de platitude du modèle à trois degrés de liberté d'un véhicule de rentrée atmosphérique, et nous calculerons les sorties plates associées par calcul formel.

## Chapitre 2

# Considérations pratiques sur quelques algorithmes de calcul des sorties plates

“Déstructurer pour reconstruire impose de ne rien  
accepter comme immuable.”

**Thierry Marx**

---

### Résumé :

*Ce chapitre est consacré aux aspects pratiques liés à l'utilisation des algorithmes de calcul des sorties plates. Nous proposons une extension des conditions nécessaires et suffisantes de platitude établies dans [7] au cas des systèmes linéaires multidimensionnels à retards. Un algorithme constructif, basé sur les outils classiques de l'algèbre linéaire polynomiale, est proposé afin de vérifier si un système linéaire à retards est  $\delta$ -plat et, le cas échéant, de calculer des sorties  $\delta$ -plates candidates.*

---

## 2.1 Introduction

La caractérisation formelle de sorties plates est fortement motivée par un besoin d'établir des lignes de conduite systématiques pouvant être utilisées lors du développement de lois de commande pour les systèmes non linéaires ayant la propriété d'être plats. Bien que des conditions nécessaires et suffisantes de platitude existent, le calcul systématique et efficace de sorties plates reste encore un problème ouvert.

Dans ce chapitre, nous nous attacherons à étendre la méthodologie de calcul de sorties plates par décompositions de Smith, développée dans [7], à la classe des systèmes linéaires différentiels à retards. Nous proposerons un algorithme constructif, basé sur les outils classiques de l'algèbre linéaire polynomiale, afin de vérifier si un système linéaire à retards est  $\delta$ -plat et, le cas échéant, de calculer une sortie  $\delta$ -plate candidate.

## 2.2 Calcul de sorties plates pour les systèmes linéaires à retards

Dans cette section, un algorithme constructif simple, dédié au calcul formel de sorties  $\delta$ -plates candidates pour les systèmes linéaires à retards, est proposé. Des approches différentes peuvent être trouvées dans [24, 25]. L'approche retenue ici est basée sur des résultats classiques provenant de l'algèbre polynomiale linéaire et, par conséquent, peut être vue comme une extension des résultats obtenus dans [7] pour les systèmes linéaires régis par des équations différentielles ordinaires. Des exemples sont fournis afin d'illustrer la méthodologie proposée. Ce travail a été réalisé en collaboration avec le professeur Jean Lévine du Centre Automatique et Systèmes (CAS) de l'École Nationale Supérieure des Mines de Paris.

Considérons un système linéaire à retards simple défini par [18]

$$\begin{cases} \dot{x}_1(t) - x_1(t - \tau) + 2x_1(t) + 2x_2(t) = 2u(t - \tau), \\ \dot{x}_1(t) + \dot{x}_2(t) = \dot{u}(t - \tau) + u(t). \end{cases} \quad (2.1)$$

En notant  $\delta$  l'opérateur de retard de durée  $\tau$ , le système (2.1) peut être représenté au moyen de matrices à éléments dans l'anneau polynomial  $\mathfrak{R}[\frac{d}{dt}, \delta]$  (voir section 1.2) des opérateurs linéaires en  $\frac{d}{dt}$  et à coefficients dans le corps  $\mathfrak{R}(\delta)$ , tel que

$$Ax = Bu, \quad (2.2)$$

où  $x = (x_1, x_2)^T$  désigne l'état du système,  $u$  la commande, et où les matrices  $A \in \mathcal{M}_{2,2}(\frac{d}{dt}, \delta)$  et  $B \in \mathcal{M}_{2,1}(\frac{d}{dt}, \delta)$  sont définies respectivement par

$$A = \begin{pmatrix} \frac{d}{dt} - \delta + 2 & 2 \\ \frac{d}{dt} & \frac{d}{dt} \end{pmatrix}, \quad B = \begin{pmatrix} 2\delta \\ \frac{d}{dt}\delta + 1 \end{pmatrix}. \quad (2.3)$$

Il a été montré dans [18] que le système (2.1) est  $\delta$ -plat, avec  $z = x_1$  comme sortie  $\delta$ -plate. Ainsi, l'état et la commande du système peuvent être réécrits en fonction de  $z$ , de ses

dérivées et d'un nombre fini de retards sous la forme

$$\begin{aligned} x_1(t) &= z(t), \\ x_2(t) &= -\frac{1}{2}\ddot{z}(t - \tau) + \frac{1}{2}\dot{z}(t - 2\tau) - \frac{1}{2}\dot{z}(t) + \frac{1}{2}z(t - \tau) - z(t), \end{aligned}$$

et

$$u(t) = -\frac{1}{2}\ddot{z}(t) + \frac{1}{2}\dot{z}(t - \tau). \quad (2.4)$$

Un calcul systématique de sorties  $\delta$ -plates candidates pour les systèmes linéaires à retards peut se révéler être une problématique intéressante dans le cadre de la théorie de la platitude. Ainsi, l'objectif de cette section est de fournir un algorithme constructif simple permettant de vérifier si un système à retards est  $\delta$ -libre et, le cas échéant, de calculer une sortie  $\delta$ -plate candidate, ainsi que les expressions des variables du système en fonction de cette sortie  $\delta$ -plate, en utilisant un nombre fini de différentiations, de retards et d'avances.

Afin de caractériser les sorties  $\delta$ -plates pour les systèmes linéaires à retards, nous allons utiliser un outil important provenant de l'algèbre polynomiale linéaire, permettant de transformer une matrice en sa forme normale de Smith (voir annexe A). Cependant, la forme de Smith existant seulement pour les matrices définies sur un anneau principal (PID), leurs éléments polynomiaux peuvent être redéfinis sur l'anneau  $\mathfrak{K}(\delta)[\frac{d}{dt}]$  des polynômes en  $\frac{d}{dt}$  à coefficients dans le corps  $\delta$ -différentiel  $\mathfrak{K}(\delta)$ , au lieu de l'anneau  $\mathfrak{K}[\frac{d}{dt}, \delta]$  qui n'est pas euclidien. Dans la suite,  $\mathcal{M}_{p,q}(\frac{d}{dt}, \delta)$  désigne le module des matrices de taille  $p \times q$  définies sur l'anneau  $\mathfrak{K}(\delta)[\frac{d}{dt}]$ , i.e., les matrices polynomiales en  $\frac{d}{dt}$  à coefficients rationnels en  $\delta$  et méromorphiques en  $t$ .

### 2.2.1 Résultat principal

Dans cette section, nous proposons un algorithme constructif simple permettant de calculer des sorties  $\delta$ -plates pour les systèmes linéaires à retards. Le système (1.6) est considéré sous forme implicite de manière à tenir compte de sa propriété naturelle d'invariance par bouclage dynamique endogène [31]. Par conséquent, la sortie  $\delta$ -plate (1.7) est recherchée sous la forme d'une combinaison linéaire de l'état du système, de ses dérivées et d'un nombre fini de retards, i.e.,  $z = Px$  avec  $P \in \mathcal{M}_{m,n}(\frac{d}{dt}, \delta)$ . En outre, les expressions de l'état et des commandes définies respectivement par (1.8) et (1.9) peuvent être réécrites sous la forme de matrices polynomiales telles que  $x = Qz$ ,  $Q \in \mathcal{M}_{n,m}(\frac{d}{dt}, \delta)$ , et  $u = Rz$ ,  $R \in \mathcal{M}_{m,m}(\frac{d}{dt}, \delta)$ . Les matrices  $P$ ,  $Q$  et  $R$  sont appelées *opérateurs de définition* de la sortie  $\delta$ -plate  $z$ .

Le théorème suivant fournit un résultat constructif permettant de calculer les opérateurs de définition  $P$ ,  $Q$  et  $R$  sous des environnements de calcul formel standards.

**Théorème 2.1.** *Considérons le système linéaire à retards défini par  $Ax = Bu$ , où  $A \in \mathcal{M}_{n,n}(\frac{d}{dt}, \delta)$  et  $B \in \mathcal{M}_{n,m}(\frac{d}{dt}, \delta)$ . Le système à retards est  $\delta$ -plat si et seulement s'il existe un ensemble d'opérateurs de définition  $P$ ,  $Q$ , et  $R$  satisfaisant  $z = Px$ ,  $x = Qz$  et  $u = Rz$ ,*



tels que

$$\begin{aligned} Q &= \tilde{Q}(0_{n-m,m}, I_m)^T, \text{ où } \tilde{Q} \in \text{D-Smith}((0_{n-m,m}, I_{n-m})MA), \text{ et } M \in \text{G-Smith}(B), \\ R &= (I_m, 0_{m,n-m})MAQ, \\ P &= (I_m, 0_{m,n-m})\tilde{P}, \text{ où } \tilde{P} \in \text{G-Smith}(Q). \end{aligned}$$

Dans la perspective de l'implémentation du Théorème 2.1 sous un environnement de calcul formel, sa formulation algorithmique est donnée ci-après (voir l'algorithme 2).

**Algorithme 2** : Caractérisation algébrique d'un système linéaire  $\delta$ -plat.

**Entrées** : Deux matrices  $A \in \mathcal{M}_{n,n}(\frac{d}{dt}, \delta)$  et  $B \in \mathcal{M}_{n,m}(\frac{d}{dt}, \delta)$ .

**Sorties** : Un indicateur de défaut de platitude  $d$  et un ensemble d'opérateurs de définition  $P \in \mathcal{M}_{m,n}(\frac{d}{dt}, \delta)$ ,  $Q \in \mathcal{M}_{n,m}(\frac{d}{dt}, \delta)$  et  $R \in \mathcal{M}_{m,m}(\frac{d}{dt}, \delta)$ , vérifiant  $z = Px$ ,  $x = Qz$  et  $u = Rz$ .

**Initialisation** :  $d = 0$ .

**Algorithme** :

- 1 Calculer une décomposition de Smith de  $B$  telle que  $MBN = (I_m, 0_{n-m,m})^T$ , et poser  $F = (0_{n-m,m}, I_{n-m})MA$ .
- 2 Calculer une décomposition de Smith de la matrice  $F$  telle que  $VF\tilde{Q} = (I_{n-m}, 0_{n-m,m})$ . Si une telle décomposition n'existe pas, le système n'est pas  $\delta$ -plat et  $d$  est mis à 1.
- 3 Sinon, calculer la matrice  $Q$  en utilisant la relation  $Q = \tilde{Q}(0_{n-m,m}, I_m)^T$ .
- 4 Calculer la matrice  $R$  en utilisant la relation  $R = (I_m, 0_{m,n-m})MAQ$ .
- 5 Calculer une décomposition de Smith de la matrice  $Q$  telle que  $\tilde{P}QU = (I_m, 0_{n-m,m})^T$ .
- 6 Sinon, calculer la matrice  $P$  en utilisant la relation  $P = (I_m, 0_{m,n-m})\tilde{P}$ .

## 2.2.2 Exemples

### Exemple introductif

Considérons de nouveau le système à retards simple avec une commande et deux états défini par (2.1) et sa forme matricielle polynomiale donnée par (2.2), où les matrices  $A$  et  $B$  de la forme explicite sont définies par (2.3).

Afin de mettre le système (2.2) sous forme implicite, commençons par calculer une décomposition de Smith à gauche de la matrice  $B$ . En la multipliant à gauche par la matrice  $M = M_3M_2M_1$ ,  $M \in \mathcal{U}_2(\frac{d}{dt}, \delta)$ , dont les éléments sont donnés par

$$M_1 = \begin{pmatrix} 1 & 0 \\ -\frac{1}{2}\frac{d}{dt} & 1 \end{pmatrix}, \quad M_2 = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \quad M_3 = \begin{pmatrix} 1 & 0 \\ -2\delta & 1 \end{pmatrix},$$

on obtient

$$MB = \begin{pmatrix} -\frac{1}{2}\frac{d}{dt} & 1 \\ \frac{d}{dt}\delta + 1 & -2\delta \end{pmatrix} \begin{pmatrix} 2\delta \\ \frac{d}{dt}\delta + 1 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \quad (2.5)$$

Ensuite, en posant  $\widetilde{M} = (0 \ 1)M$ , et en utilisant la décomposition de Smith (2.5), la forme implicite associée à (2.2) est donnée par

$$\widetilde{M}A = \left( \frac{d^2}{dt^2}\delta - \frac{d}{dt}\delta^2 + \frac{d}{dt} - \delta + 2 \quad 2 \right),$$

à laquelle correspond l'équation différentielle implicite

$$\ddot{x}_1(t - \tau) - \dot{x}_1(t - 2\tau) + \dot{x}_1(t) - x_1(t - \tau) + 2x_1(t) + 2x_2(t) = 0.$$

A présent, calculons une décomposition de Smith à droite de la matrice  $\widetilde{M}A$ . Nous obtenons les matrices de transformation unimodulaires définies par

$$\widetilde{Q}_1 = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \quad \widetilde{Q}_2 = \begin{pmatrix} \frac{1}{2} & -\frac{1}{2}\left(\frac{d^2}{dt^2}\delta - \frac{d}{dt}\delta^2 + \frac{d}{dt} - \delta + 2\right) \\ 0 & 1 \end{pmatrix}.$$

La matrice de transformation unimodulaire résultante  $\widetilde{Q} \in \mathcal{U}_2\left(\frac{d}{dt}, \delta\right)$  est alors donnée par

$$\widetilde{Q} = \widetilde{Q}_1\widetilde{Q}_2 = \begin{pmatrix} 0 & 1 \\ \frac{1}{2} & -\frac{1}{2}\left(\frac{d^2}{dt^2}\delta - \frac{d}{dt}\delta^2 + \frac{d}{dt} - \delta + 2\right) \end{pmatrix},$$

telle que  $\widetilde{M}A\widetilde{Q} = (1 \ 0)$ . Il est alors possible d'obtenir la matrice reliant les entrées de commande à la sortie  $\delta$ -plate candidate<sup>11</sup>  $z$  telle que

$$R = (1 \ 0)MAQ = \left( -\frac{1}{2}\frac{d^2}{dt^2} + \frac{1}{2}\frac{d}{dt}\delta \right),$$

où  $Q = \widetilde{Q}(0 \ 1)^T$ . Par conséquent, l'expression de l'entrée de commande en fonction de  $z$ , de ses dérivées et du retard  $\tau$ , est donnée par

$$u(t) = -\frac{1}{2}\ddot{z}(t) + \frac{1}{2}\dot{z}(t - \tau).$$

De plus, l'état du système (2.1) peut être déterminé en utilisant la relation  $x = Qz$ , telle que

$$\begin{aligned} \begin{pmatrix} x_1(t) \\ x_2(t) \end{pmatrix} &= \begin{pmatrix} 1 \\ -\frac{1}{2}\left(\frac{d^2}{dt^2}\delta - \frac{d}{dt}\delta^2 + \frac{d}{dt} - \delta + 2\right) \end{pmatrix} z(t) \\ &= \begin{pmatrix} z(t) \\ -\frac{1}{2}\ddot{z}(t - \tau) + \frac{1}{2}\dot{z}(t - 2\tau) - \frac{1}{2}\dot{z}(t) + \frac{1}{2}z(t - \tau) - z(t) \end{pmatrix}. \end{aligned}$$

Afin de caractériser la sortie  $\delta$ -plate, une inverse à gauche de la matrice  $Q$  doit être calculée. Ceci peut être réalisé en calculant une décomposition de Smith à gauche  $\widetilde{P}QU = (1, 0)^T$  de  $Q$ , telle que

$$\begin{pmatrix} 1 & 0 \\ \frac{1}{2}\left(\frac{d^2}{dt^2}\delta - \frac{d}{dt}\delta^2 + \frac{d}{dt} - \delta + 2\right) & 1 \end{pmatrix} \begin{pmatrix} 1 \\ -\frac{1}{2}\left(\frac{d^2}{dt^2}\delta - \frac{d}{dt}\delta^2 + \frac{d}{dt} - \delta + 2\right) \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \end{pmatrix},$$

avec  $U = 1$ . Finalement, la sortie  $\delta$ -plate est donnée par  $z = Px = (1 \ 0)\widetilde{P}x$ , c'est à dire  $z = x_1$ .

<sup>11</sup>l'expression de la sortie  $\delta$ -plate n'est pas encore connue à ce niveau du processus.

### Un exemple plus général

Considérons le système linéaire à retard suivant, comprenant un opérateur de retard  $\delta$  de durée  $\tau$ , dont la forme explicite est donnée par

$$\left\{ \begin{array}{l} \dot{x}_1(t) + x_1^{(2)}(t) - 2x_1^{(2)}(t - \tau) + x_1^{(3)}(t) + x_1^{(4)}(t - \tau) - x_2^{(3)}(t) + x_2^{(5)}(t) - x_3^{(2)}(t) \\ -\dot{x}_4(t) + x_4^{(3)}(t) = u_1(t) + \dot{u}_1(t) + u_2(t), \\ \dot{x}_1(t) + x_1^{(2)}(t) - 2x_1^{(2)}(t - \tau) + x_1^{(3)}(t) + x_1^{(4)}(t - \tau) - x_2^{(3)}(t) + x_2^{(5)}(t) - x_3^{(2)}(t) \\ -\dot{x}_4(t) + x_4^{(3)}(t) = u_1(t) + \dot{u}_1(t) + u_2(t), \\ \dot{x}_1(t) + \dot{x}_1(t - \tau) - \dot{x}_1(t - 2\tau) + x_1^{(2)}(t) + x_1^{(2)}(t - \tau) + x_1^{(2)}(t - 2\tau) - x_1^{(3)}(t - \tau) \\ + 2\dot{x}_2(t) + \dot{x}_2(t - \tau) - x_2^{(2)}(t) - x_2^{(4)}(t) + \dot{x}_3(t) + x_3^{(2)}(t - \tau) - x_4(t) - x_4(t - \tau) \\ - x_4^{(2)}(t) = u_1^{(2)}(t - \tau) + \dot{u}_2(t - \tau), \\ -x_1(t - 2\tau) + \dot{x}_1(t - 3\tau) + x_1^{(2)}(t - 2\tau) - x_2(t - \tau) + \dot{x}_2(t - 2\tau) + x_2^{(3)}(t - \tau) \\ - x_3(t - \tau) + \dot{x}_3(t - 2\tau) + \dot{x}_4(t - \tau) = \dot{u}_1(t - 2\tau) + u_2(t - 2\tau), \\ \dot{x}_1(t - \tau) + \dot{x}_2(t) + \dot{x}_3(t) = \dot{u}_1(t) + u_2. \\ \dot{x}_1(t) + \dot{x}_1(t - \tau) - \dot{x}_1(t - 2\tau) + x_1^{(2)}(t) + x_1^{(2)}(t - \tau) + x_1^{(2)}(t - 2\tau) - x_1^{(3)}(t - \tau) \\ + 2\dot{x}_2(t) + \dot{x}_2(t - \tau) - x_2^{(2)}(t) - x_2^{(4)}(t) + \dot{x}_3(t) + x_3^{(2)}(t - \tau) - x_4(t) - x_4(t - \tau) \\ - x_4^{(2)}(t) = u_1^{(2)}(t - \tau) + \dot{u}_2(t - \tau), \\ -x_1(t - 2\tau) + \dot{x}_1(t - 3\tau) + x_1^{(2)}(t - 2\tau) - x_2(t - \tau) + \dot{x}_2(t - 2\tau) + x_2^{(3)}(t - \tau) \\ - x_3(t - \tau) + \dot{x}_3(t - 2\tau) + \dot{x}_4(t - \tau) = \dot{u}_1(t - 2\tau) + u_2(t - 2\tau), \\ \dot{x}_1(t - \tau) + \dot{x}_2(t) + \dot{x}_3(t) = \dot{u}_1(t) + u_2. \end{array} \right. \quad (2.6)$$

En posant  $x = (x_1, x_2, x_3, x_4)^T$  l'état,  $u = (u_1, u_2)^T$  le vecteur de commandes, et  $\delta$  l'opérateur de retard de durée  $\tau$ , le système (2.6) peut être réécrit sous la forme  $Ax = Bu$ , où les matrices  $A \in \mathcal{M}_{4,4}(\frac{d}{dt}, \delta)$  et  $B \in \mathcal{M}_{4,2}(\frac{d}{dt}, \delta)$  sont définies respectivement par

$$A = \begin{pmatrix} \frac{d}{dt} + \frac{d^3}{dt^3} + \frac{d^2}{dt^2} - 2\frac{d^2}{dt^2}\delta + \frac{d^4}{dt^4}\delta & -\frac{d^3}{dt^3} + \frac{d^5}{dt^5} & \dots \\ \frac{d^2}{dt^2} + \frac{d}{dt} + \frac{d}{dt}\delta + \frac{d^2}{dt^2}\delta - \frac{d}{dt}\delta^2 - \frac{d^3}{dt^3}\delta + \frac{d^2}{dt^2}\delta^2 & 2\frac{d}{dt} - \frac{d^2}{dt^2} + \frac{d}{dt}\delta - \frac{d^4}{dt^4} & \dots \\ \frac{d^2}{dt^2}\delta^2 - \delta^2 + \frac{d}{dt}\delta^3 & -\delta + \frac{d^3}{dt^3}\delta + \frac{d}{dt}\delta^2 & \dots \\ \frac{d}{dt}\delta & \frac{d}{dt} & \dots \\ \dots & -\frac{d^2}{dt^2} & -\frac{d}{dt} + \frac{d^3}{dt^3} \\ \dots & \frac{d}{dt} + \frac{d^2}{dt^2}\delta & -1 - \delta - \frac{d^2}{dt^2} \\ \dots & -\delta + \frac{d}{dt}\delta^2 & \frac{d}{dt}\delta \\ \dots & \frac{d}{dt} & 0 \end{pmatrix}$$

et

$$B = \begin{pmatrix} \frac{d}{dt} + 1 & 1 \\ \frac{d^2}{dt^2}\delta & \frac{d}{dt}\delta \\ \frac{d}{dt}\delta^2 & \delta^2 \\ \frac{d}{dt} & 1 \end{pmatrix}.$$

Afin de mettre le système (2.6) sous forme implicite, calculons une décomposition de Smith à gauche de  $B$ . En multipliant la matrice  $B$  à gauche par la matrice de transformation

unimodulaire  $M = M_4 M_3 M_2 M_1$ ,  $M \in \mathcal{U}_4(\frac{d}{dt}, \delta)$ , dont les éléments sont donnés par

$$M_1 = \begin{pmatrix} 1 & 0 & 0 & -1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \quad M_2 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ -\frac{d^2}{dt^2}\delta & 1 & 0 & 0 \\ -\frac{d}{dt}\delta^2 & 0 & 1 & 0 \\ -\frac{d}{dt} & 0 & 0 & 1 \end{pmatrix},$$

$$M_3 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}, \quad M_4 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & -\frac{d}{dt}\delta & 1 & 0 \\ 0 & -\delta^2 & 0 & 1 \end{pmatrix},$$

on obtient la décomposition de Smith

$$MB = \begin{pmatrix} 1 & 0 & 0 & -1 \\ -\frac{d}{dt} & 0 & 0 & \frac{d}{dt} + 1 \\ 0 & 1 & 0 & -\frac{d}{dt}\delta \\ 0 & 0 & 1 & -\delta^2 \end{pmatrix} \begin{pmatrix} \frac{d}{dt} + 1 & 1 \\ \frac{d^2}{dt^2}\delta & \frac{d}{dt}\delta \\ \frac{d}{dt}\delta^2 & \delta^2 \\ \frac{d}{dt} & 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \\ 0 & 0 \end{pmatrix},$$

à partir de laquelle, en posant  $\widetilde{M} = (0_{2,2} \quad I_2)M$ , on déduit la forme implicite

$$\widetilde{M}A = \begin{pmatrix} \frac{d^2}{dt^2} + \frac{d}{dt} + \frac{d}{dt}\delta + \frac{d^2}{dt^2}\delta - \frac{d}{dt}\delta^2 - \frac{d^3}{dt^3}\delta & 2\frac{d}{dt} - \frac{d^2}{dt^2} + \frac{d}{dt}\delta - \frac{d^4}{dt^4} - \frac{d^2}{dt^2}\delta & \dots \\ \frac{d^2}{dt^2}\delta^2 - \delta^2 & -\delta + \frac{d^3}{dt^3}\delta & \dots \\ \dots & \frac{d}{dt} & -1 - \delta - \frac{d^2}{dt^2} \\ \dots & -\delta & \frac{d}{dt}\delta & \frac{d^2}{dt^2} \end{pmatrix},$$

à laquelle correspond le système d'équations différentielles retardées

$$\begin{cases} \dot{x}_1(t) + \dot{x}_1(t - \tau) - \dot{x}_1(t - 2\tau) + x_1^{(2)}(t) + x_1^{(2)}(t - \tau) - x_1^{(3)}(t - \tau) + 2\dot{x}_2(t) + \dot{x}_2(t - \tau) \\ -x_2^{(2)}(t) - x_2^{(2)}(t - \tau) - x_2^{(4)}(t) + \dot{x}_3(t) - x_4(t) - x_4(t - \tau) - x_4^{(2)}(t) = 0, \\ -x_1(t - 2\tau) + x_1^{(2)}(t - 2\tau) - x_2(t - \tau) + x_2^{(3)}(t - \tau) - x_3(t - \tau) + \dot{x}_4(t - \tau) = 0. \end{cases}$$

Maintenant, calculons une décomposition de Smith à droite de  $\widetilde{M}A$ . Les matrices de transformation unimodulaires associées sont définies par

$$\widetilde{Q}_1 = \begin{pmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ \frac{d}{dt} & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix},$$

$$\widetilde{Q}_2 = \begin{pmatrix} -(1 + \delta)^{-1} & (1 + \delta)^{-1} \frac{d}{dt} & (1 + \delta)^{-1} \frac{d}{dt} \left(1 - \frac{d^3}{dt^3}\right) + \frac{d}{dt} \left(1 - \frac{d}{dt}\right) & \dots \\ 0 & 1 & 0 & \dots \\ 0 & 0 & 1 & \dots \\ 0 & 0 & 0 & \dots \\ \dots & -(1 + \delta)^{-1} \frac{d}{dt}\delta \left(\delta + \frac{d^2}{dt^2}\right) & & \\ \dots & 0 & & \\ \dots & 0 & & \\ \dots & 1 & & \end{pmatrix},$$

$$\tilde{Q}_3 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & -\delta^{-1} & -1 + \frac{d^3}{dt^3} & \frac{d^2}{dt^2}\delta - \delta \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}.$$

La matrice de transformation unimodulaire à droite  $\tilde{Q} \in \mathcal{U}_4(\frac{d}{dt}, \delta)$  est alors calculée par

$$\tilde{Q} = \tilde{Q}_1 \tilde{Q}_2 \tilde{Q}_3 = \begin{pmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ -(1+\delta)^{-1} \frac{d}{dt} & -\delta^{-1} (1+\delta)^{-1} \left(1 + \delta + \frac{d^2}{dt^2}\right) & \frac{d^2}{dt^2} - 1 & \frac{d^3}{dt^3} + \frac{d^2}{dt^2} - \delta \\ -(1+\delta)^{-1} & -\delta^{-1} (1+\delta)^{-1} \frac{d}{dt} & \left(1 - \frac{d}{dt}\right) \frac{d}{dt} & \left(\frac{d}{dt} + 1 - \delta\right) \frac{d}{dt} \end{pmatrix},$$

telle que

$$\tilde{M}A\tilde{Q} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix} \quad (2.7)$$

**Remarque 2.1.** De par l'utilisation de l'opérateur  $\delta^{-1}$  dans la décomposition de Smith, le module associé n'est pas libre.

Nous déduisons que la matrice reliant les entrées de commande du système (2.6) à la sortie  $\delta$ -plate est définie par

$$R = (I_2 \quad 0_{2,2})MAQ = \begin{pmatrix} -\frac{d^3}{dt^3} & \frac{d}{dt} - \frac{d^4}{dt^4} - \frac{d^3}{dt^3} \\ \left(\frac{d}{dt} + 1\right) \frac{d^3}{dt^3} & -\frac{d^2}{dt^2} + \frac{d^5}{dt^5} + 2\frac{d^4}{dt^4} + \frac{d^3}{dt^3} \end{pmatrix},$$

où  $Q = \tilde{Q}(0_{2,2} \quad I_2)^T$ . Ainsi, l'expression des commandes en fonction de la sortie  $\delta$ -plate et de ses dérivées est donnée par

$$\begin{aligned} \begin{pmatrix} u_1(t) \\ u_2(t) \end{pmatrix} &= \begin{pmatrix} -\frac{d^3}{dt^3} & \frac{d}{dt} - \frac{d^4}{dt^4} - \frac{d^3}{dt^3} \\ \left(\frac{d}{dt} + 1\right) \frac{d^3}{dt^3} & -\frac{d^2}{dt^2} + \frac{d^5}{dt^5} + 2\frac{d^4}{dt^4} + \frac{d^3}{dt^3} \end{pmatrix} \begin{pmatrix} z_1(t) \\ z_2(t) \end{pmatrix} \\ &= \begin{pmatrix} -z_1^{(3)}(t) + \dot{z}_2(t) - z_2^{(3)}(t) - z_2^{(4)}(t) \\ z_1^{(3)}(t) + z_1^{(4)}(t) - z_2^{(2)}(t) + z_2^{(3)}(t) + 2z_2^{(4)}(t) + z_2^{(5)}(t) \end{pmatrix}. \end{aligned}$$

En outre, l'état du système (2.6) peut être obtenu grâce à la relation  $x = Qz$ , telle que

$$\begin{aligned} \begin{pmatrix} x_1(t) \\ x_2(t) \\ x_3(t) \\ x_4(t) \end{pmatrix} &= \begin{pmatrix} 0 & 1 \\ 1 & 0 \\ \frac{d^2}{dt^2} - 1 & \frac{d^3}{dt^3} + \frac{d^2}{dt^2} - \delta \\ \left(1 - \frac{d}{dt}\right) \frac{d}{dt} & \left(\frac{d}{dt} + 1 - \delta\right) \frac{d}{dt} \end{pmatrix} \begin{pmatrix} z_1(t) \\ z_2(t) \end{pmatrix} \\ &= \begin{pmatrix} z_2(t) \\ z_1(t) \\ -z_1(t) + z_1^{(2)}(t) - z_2(t - \tau) + z_2^{(2)}(t) + z_2^{(3)}(t) \\ \dot{z}_1(t) - z_1^{(2)}(t) + \dot{z}_2(t) - \dot{z}_2(t - \tau) + z_2^{(2)}(t) \end{pmatrix}. \end{aligned}$$

Finalement, calculons une inverse à gauche  $\tilde{P}$  de la matrice  $Q$  afin d'obtenir la matrice reliant la sortie  $\delta$ -plate à l'état du système. Ceci peut être réalisé en calculant une décomposition de Smith à gauche de  $Q$ , telle que  $\tilde{P}QU = (I_2 \ 0_{2,2})^T$ . Les matrices élémentaires de transformation à gauche sont alors définies par

$$\tilde{P}_1 = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \quad \tilde{P}_2 = \begin{pmatrix} 1 & & 0 & 0 & 0 \\ 0 & 1 & & 0 & 0 \\ -\frac{d^2}{dt^2} + 1 & -\frac{d^3}{dt^3} - \frac{d^2}{dt^2} + \delta & 1 & 0 & 0 \\ -\left(1 - \frac{d}{dt}\right) \frac{d}{dt} & -\left(\frac{d}{dt} + 1 - \delta\right) \frac{d}{dt} & 0 & 1 & 0 \end{pmatrix}, \quad U = I_2.$$

On en déduit que la matrice de transformation unimodulaire résultante  $\tilde{P} \in \mathcal{U}_4(\frac{d}{dt}, \delta)$  est définie par

$$\tilde{P} = \tilde{P}_2 \tilde{P}_1 = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ -\frac{d^3}{dt^3} - \frac{d^2}{dt^2} + \delta & -\frac{d^2}{dt^2} + 1 & 1 & 0 \\ \frac{d\delta}{dt} - \frac{d^2}{dt^2} - \frac{d}{dt} & -\frac{d}{dt} + \frac{d^2}{dt^2} & 0 & 1 \end{pmatrix}.$$

Finalement, la sortie  $\delta$ -plate est calculée en utilisant la relation  $z = Px = (I_2 \ 0_{2,2}) \tilde{P}x$ , telle que  $z_1 = x_2$  et  $z_2 = x_1$ .

### Corde vibrante avec une masse

Considérons à nouveau le modèle de corde vibrante avec une masse (1.27) décrit dans la section 1.2. En notant  $x = (\psi_1, \phi_1, \psi_2, \phi_2)^T$  l'état,  $(u, v)$  le vecteur des commandes, et  $\sigma_1, \sigma_2$  les opérateurs de retard de durées respectives  $\mu_1 L_1$  et  $\mu_2 L_2$ , le système (1.27) peut être réécrit sous la forme  $Ax = Bu$ , avec

$$A = \begin{pmatrix} 1 & 1 & -1 & -1 \\ \frac{d}{dt} + \eta_1 & \frac{d}{dt} - \eta_1 & \eta_2 & -\eta_2 \\ \sigma_1^{-1} & \sigma_1 & 0 & 0 \\ 0 & 0 & \sigma_2 & \sigma_2^{-1} \end{pmatrix}, \quad A \in \mathcal{M}_{4,4}(\frac{d}{dt}, \delta),$$

et

$$B = \begin{pmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{pmatrix}, \quad B \in \mathcal{M}_{4,2}(\frac{d}{dt}, \delta).$$

La forme implicite associée est alors obtenue en définissant la matrice

$$M = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}, \quad M \in \mathcal{U}_4(\frac{d}{dt}, \delta),$$

telle que

$$\tilde{M}A = \begin{pmatrix} 1 & 1 & -1 & -1 \\ \frac{d}{dt} + \eta_1 & \frac{d}{dt} - \eta_1 & \eta_2 & -\eta_2 \end{pmatrix},$$

où  $\widetilde{M}$  est définie par  $\widetilde{M} = (0_{2,2} \quad I_2)M$ .

Ensuite, une décomposition de Smith à droite de la matrice  $\widetilde{M}A$  est donnée par

$$W\widetilde{M}A\widetilde{Q} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix},$$

où les matrices de transformation unimodulaires  $W \in \mathcal{U}_2(\frac{d}{dt}, \delta)$  et  $\widetilde{Q} \in \mathcal{U}_4(\frac{d}{dt}, \delta)$  sont définies respectivement par

$$W = \begin{pmatrix} 1 & 0 \\ -\frac{d}{dt} - \eta_1 & 1 \end{pmatrix},$$

$$\widetilde{Q}_1 = \begin{pmatrix} 1 & -1 & 1 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix},$$

et

$$\widetilde{Q}_2 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & -\frac{1}{2\eta_1} & \frac{1}{2\eta_1} \frac{d}{dt} + \frac{\eta_1 + \eta_2}{2\eta_1} & \frac{1}{2\eta_1} \frac{d}{dt} + \frac{\eta_1 - \eta_2}{2\eta_1} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix},$$

telles que

$$\widetilde{Q} = \widetilde{Q}_1 \widetilde{Q}_2 = \begin{pmatrix} 1 & \frac{1}{2\eta_1} & -\frac{1}{2\eta_1} \frac{d}{dt} + \frac{\eta_1 - \eta_2}{2\eta_1} & -\frac{1}{2\eta_1} \frac{d}{dt} + \frac{\eta_1 + \eta_2}{2\eta_1} \\ 0 & -\frac{1}{2\eta_1} & \frac{1}{2\eta_1} \frac{d}{dt} + \frac{\eta_1 + \eta_2}{2\eta_1} & \frac{1}{2\eta_1} \frac{d}{dt} + \frac{\eta_1 - \eta_2}{2\eta_1} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}.$$

La matrice reliant les entrées de commande  $u$  et  $v$  à la sortie  $\delta$ -plate candidate  $(z_1, z_2)$  et aux opérateurs de retard  $(\sigma_1, \sigma_2)$  et d'avance  $(\sigma_1^{-1}, \sigma_2^{-1})$ , est donnée par

$$R = (I_2 \quad 0_{2,2})MAQ = \begin{pmatrix} \frac{1}{2\eta_1} \frac{d}{dt} \sigma_1 - \frac{1}{2\eta_1} \frac{d}{dt} \sigma_1^{-1} + \frac{\eta_1 + \eta_2}{2\eta_1} \sigma_1 + \frac{\eta_1 - \eta_2}{2\eta_1} \sigma_1^{-1} & \sigma_2 \\ \frac{1}{2\eta_1} \frac{d}{dt} \sigma_1 - \frac{1}{2\eta_1} \frac{d}{dt} \sigma_1^{-1} + \frac{\eta_1 - \eta_2}{2\eta_1} \sigma_1 + \frac{\eta_1 + \eta_2}{2\eta_1} \sigma_1^{-1} & \sigma_2^{-1} \end{pmatrix}^T,$$

avec  $Q = \widetilde{Q}(0_{2,2} \quad I_2)^T$ . Par conséquent, les commandes du système peuvent s'exprimer en fonction de la sortie  $\delta$ -plate, de ses dérivées temporelles, et d'un nombre fini de retards et d'avances, telles que

$$\begin{aligned} u(t) &= \frac{1}{2\eta_1} [\dot{z}_1(t - \mu_1 L_1) - \dot{z}_1(t + \mu_1 L_1) + \dot{z}_2(t - \mu_1 L_1) - \dot{z}_2(t + \mu_1 L_1) \\ &\quad + (\eta_1 + \eta_2)(z_1(t - \mu_1 L_1) + z_2(t + \mu_1 L_1)) + (\eta_1 - \eta_2)(z_1(t + \mu_1 L_1) + z_2(t - \mu_1 L_1))], \\ v(t) &= z_1(t - \mu_2 L_2) + z_2(t + \mu_2 L_2). \end{aligned}$$

De plus, l'état du système peut être déduit de la relation  $x = Qz$ , telle que

$$\begin{aligned} \begin{pmatrix} \psi_1(t) \\ \phi_1(t) \\ \psi_2(t) \\ \phi_2(t) \end{pmatrix} &= \begin{pmatrix} -\frac{1}{2\eta_1} \frac{d}{dt} + \frac{\eta_1 - \eta_2}{2\eta_1} & -\frac{1}{2\eta_1} \frac{d}{dt} + \frac{\eta_1 + \eta_2}{2\eta_1} \\ \frac{1}{2\eta_1} \frac{d}{dt} + \frac{\eta_1 + \eta_2}{2\eta_1} & \frac{1}{2\eta_1} \frac{d}{dt} + \frac{\eta_1 - \eta_2}{2\eta_1} \\ 1 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} z_1(t) \\ z_2(t) \end{pmatrix} \\ &= \begin{pmatrix} \frac{1}{2\eta_1} (-\dot{z}_1(t) - \dot{z}_2(t) + (\eta_1 - \eta_2)z_1(t) + (\eta_1 + \eta_2)z_2(t)) \\ \frac{1}{2\eta_1} (\dot{z}_1(t) + \dot{z}_2(t) + (\eta_1 + \eta_2)z_1(t) + (\eta_1 - \eta_2)z_2(t)) \\ z_1(t) \\ z_2(t) \end{pmatrix}. \end{aligned}$$

Ensuite, une inverse à gauche de la matrice  $Q$  peut être obtenue en calculant de nouveau une décomposition de Smith à gauche de  $Q$  telle que  $\tilde{P}QU = (I_2 \ 0_{2,2})^T$ . Les matrices de transformation unimodulaires associées sont définies par

$$\begin{aligned} \tilde{P}_1 &= \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, & \tilde{P}_2 &= \begin{pmatrix} 1 & 0 & 0 & 0 \\ -\frac{1}{2\eta_1} \frac{d}{dt} - \frac{\eta_1 + \eta_2}{2\eta_1} & 1 & 0 & 0 \\ \frac{1}{2\eta_1} \frac{d}{dt} - \frac{\eta_1 - \eta_2}{2\eta_1} & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \\ \tilde{P}_3 &= \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}, & \tilde{P}_4 &= \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & \frac{1}{2\eta_1} \frac{d}{dt} - \frac{\eta_1 + \eta_2}{2\eta_1} & 1 & 0 \\ 0 & -\frac{1}{2\eta_1} \frac{d}{dt} - \frac{\eta_1 - \eta_2}{2\eta_1} & 0 & 1 \end{pmatrix}, \end{aligned}$$

avec  $U = I_2$ , ce qui conduit à la matrice de transformation unimodulaire résultante  $\tilde{P} \in \mathcal{U}_4(\frac{d}{dt}, \delta)$  définie par

$$\tilde{P} = \tilde{P}_4 \tilde{P}_3 \tilde{P}_2 \tilde{P}_1 = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & \frac{1}{2\eta_1} \frac{d}{dt} - \frac{\eta_1 - \eta_2}{2\eta_1} & \frac{1}{2\eta_1} \frac{d}{dt} - \frac{\eta_1 + \eta_2}{2\eta_1} \\ 0 & 1 & -\frac{1}{2\eta_1} \frac{d}{dt} - \frac{\eta_1 + \eta_2}{2\eta_1} & -\frac{1}{2\eta_1} \frac{d}{dt} - \frac{\eta_1 - \eta_2}{2\eta_1} \end{pmatrix}.$$

Finalement, une sortie  $\delta$ -plate candidate  $z = Px = (I_2 \ 0_{2,2}) \tilde{P}x$  est donnée par  $z_1 = \psi_2$  et  $z_2 = \phi_2$ .

**Remarque 2.2.** Dans [27], une solution plus compliquée à été obtenue :  $z_1 = \delta_1 \phi_1 - u$  et  $z_2 = \phi_1 + \psi_1$ .

**Remarque 2.3.** Pour cet exemple, le module correspondant n'est pas libre au sens de [24] (voir aussi la remarque 1.1).

## 2.3 Conclusion

Dans ce chapitre, nous avons discuté d'un certain nombre de considérations pratiques sur quelques algorithmes de calcul formel des sorties plates. Notamment, les conditions



nécessaires et suffisantes de platitude établies dans [7] ont été étendues à la classe des systèmes linéaires différentiels à retards. La démarche proposée, simple et systématique, est basée sur le calcul de formes normales de Smith. De plus, l'algorithme proposé permet de détecter la non-commandabilité ou la non-liberté lorsque le système considéré n'est pas  $\delta$ -plat.

Dans la section suivante, nous allons appliquer l'algorithme de calcul formel développé dans [16] à la détermination de sorties plates d'un modèle à trois degrés de liberté d'un véhicule de rentrée atmosphérique.

## Chapitre 3

# Application à un véhicule de rentrée atmosphérique

“Rien ne produit autant d’effet qu’une bonne platitude.  
Cela donne à tout le monde un sentiment de parenté.”

Oscar Wilde

---

### Résumé :

*Dans ce chapitre, nous allons appliquer l’algorithme de calcul formel de sorties plates au modèle à trois degrés de liberté d’un véhicule de rentrée atmosphérique.*

*Deux types de modélisations sont considérées :*

*La première est établie sous l’hypothèse Terre plate, qui est valable pour des phases de vol de durée relativement courte, telle que la phase de dissipation d’énergie en zone terminale (TAEM) ou d’atterrissage automatique (A&L). Dans ce cas, nous montrerons que la position du centre de gravité du véhicule dans un repère local lié à la piste constitue une sortie plate du modèle.*

*La seconde modélisation considérée tient compte du mouvement de rotation terrestre et de l’aplatissement des pôles. Le modèle obtenu, en coordonnées géographiques, est surtout utilisé lors de la phase de rentrée dite hypersonique. Dans ce cas, la position du centre de gravité du véhicule constitue également une sortie plate du modèle.*

---

### 3.1 Introduction

Dans ce chapitre, nous nous proposons d'appliquer les conditions nécessaires et suffisantes de platitude établies dans [16] à un modèle à trois degrés de liberté d'un véhicule de rentrée atmosphérique. Nous allons utiliser l'approche par décompositions de Smith afin de calculer des sorties plates du modèle. La propriété de platitude sera utilisée dans le cadre du développement d'une loi de guidage autonome dans la seconde partie du manuscrit, et plus particulièrement pour l'optimisation de trajectoires de rentrée atmosphérique. Nous verrons notamment que l'utilisation de la propriété de platitude permet de réduire de manière conséquente le nombre de variables de décision intervenant dans le problème de commande optimal associé, ce dernier pouvant alors être résolu en ligne.

Dans un premier temps, nous allons caractériser la propriété de platitude d'un véhicule de rentrée atmosphérique dans le cadre d'une modélisation dite "Terre plate"<sup>12</sup>, hypothèse particulièrement justifiée lors des phases finales d'une mission de rentrée atmosphérique, lorsque les segments de trajectoire sont relativement courts, notamment durant les phases de dissipation d'énergie en zone terminale (TAEM) et d'approche et d'atterrissage automatique (A&L).

Dans un second temps, nous montrerons que la propriété de platitude est conservée lorsqu'un modèle plus général du mouvement du véhicule est considéré, tenant compte de l'excentricité, ainsi que des termes de Coriolis en  $\Omega V$  et des efforts centrifuges en  $\Omega^2 R$ , provenant du mouvement de rotation terrestre. Ce modèle est généralement utilisé dans le cadre du développement de lois de guidage en phase hypersonique, où le dimensionnement de la trajectoire ne permet pas de négliger ces termes.

### 3.2 Platitude du modèle "Terre plate"

Le modèle "Terre plate" à trois degrés de liberté est largement utilisé dans la communauté astronautique pour définir le comportement d'un véhicule de rentrée atmosphérique lors des phases finales d'une mission de rentrée atmosphérique, c'est à dire durant les phases de dissipation d'énergie en zone terminale (TAEM) et d'approche & d'atterrissage automatique (A&L). Pour ces segments de trajectoire, relativement courts, le mouvement de rotation terrestre peut alors être négligé et, par conséquent, les effets de la force de Coriolis et de la force centrifuge sur la dynamique du véhicule. Ce dernier peut alors être positionné par rapport à un repère cartésien local lié au sol, souvent attaché à l'axe de piste, et associé à la projection conique conforme de Lambert. Les coordonnées relatives du véhicule par rapport à ce repère sont alors communément appelées coordonnées "Terre plate", faisant référence au système de projection utilisé. Nous en reparlerons plus en détails dans la seconde partie de ce mémoire.

Sous ces hypothèses, les équations régissant la dynamique du centre de gravité du véhi-

---

<sup>12</sup>nous verrons dans la seconde partie que cette hypothèse revient à considérer le rayon terrestre comme constant, i.e. à négliger l'aplatissement des pôles (excentricité nulle).

cule de rentrée par rapport au repère sol local sont alors données par [32] (voir également l'annexe C)

$$\dot{x} = V \cos \chi \cos \gamma, \quad (3.1)$$

$$\dot{y} = V \sin \chi \cos \gamma, \quad (3.2)$$

$$\dot{h} = V \sin \gamma, \quad (3.3)$$

$$\dot{V} = -\frac{D(\alpha, M)}{m} - g \sin \gamma, \quad (3.4)$$

$$\dot{\gamma} = \frac{1}{mV} (L(\alpha, M) \cos \mu - Y(\beta, M) \sin \mu) - \frac{g}{V} \cos \gamma, \quad (3.5)$$

$$\dot{\chi} = \frac{1}{mV \cos \gamma} (L(\alpha, M) \sin \mu + Y(\beta, M) \cos \mu), \quad (3.6)$$

où  $(x, y, h, V, \gamma, \chi)$  est l'état, comprenant respectivement la portée longitudinale, la portée latérale, l'altitude, la vitesse, la pente sur la trajectoire, et le cap. Les entrées de commande du système sont l'incidence  $\alpha$ , le dérapage  $\beta$  et l'angle de gîte  $\mu$ . La force de portance  $L$ , de traînée  $D$  et la force latérale  $Y$  dépendent des coefficients aérodynamiques en configuration lisse, tels que

$$L(\alpha, M) = \bar{q} S C_{L_0}(\alpha, M), \quad (3.7)$$

$$D(\alpha, M) = \bar{q} S C_{D_0}(\alpha, M). \quad (3.8)$$

$$Y(\beta, M) = \bar{q} S C_{Y_0}(\beta, M), \quad (3.9)$$

où  $\bar{q} = \frac{1}{2} \rho V^2$  est la pression dynamique.

On suppose de plus que l'accélération gravitationnelle  $g$  est constante, et qu'un modèle exponentiel défini par

$$\rho = \rho_0 \exp(-h/H_0), \quad (3.10)$$

est suffisant<sup>13</sup> pour décrire la variation de la densité atmosphérique en fonction de l'altitude, dans lequel  $\rho_0$  représente la densité de l'air au niveau moyen de la mer, et  $H_0$  est un facteur d'échelle atmosphérique constant ajusté pour les basses couches de l'atmosphère.

Sous ces hypothèses, et avec les notations précédentes, nous obtenons la proposition suivante.

**Proposition 3.1.** *Le modèle non linéaire à trois degrés de liberté d'un véhicule de rentrée atmosphérique, défini par*

$$\begin{cases} \dot{x} = V \cos \chi \cos \gamma, \\ \dot{y} = V \sin \chi \cos \gamma, \\ \dot{h} = V \sin \gamma, \end{cases} \quad \begin{cases} \dot{V} = -\frac{D(\alpha, M)}{m} - g \sin \gamma, \\ \dot{\gamma} = \frac{1}{mV} (L(\alpha, M) \cos \mu - Y(\beta, M) \sin \mu) - \frac{g}{V} \cos \gamma, \\ \dot{\chi} = \frac{1}{mV \cos \gamma} (L(\alpha, M) \sin \mu + Y(\beta, M) \cos \mu), \end{cases}$$

est plat, et la sortie plate est constituée des coordonnées  $(x, y)$  de la projection conique conforme de Lambert et de l'altitude  $h$ .

<sup>13</sup>c'est souvent le cas en pratique.

**Remarque 3.1.** A première vue, il semble évident que la position du centre de masse du véhicule constitue une sortie plate du modèle à trois degrés de liberté, dès lors que l'on maîtrise de façon indépendante les trois efforts appliqués dans les trois directions du véhicule (gouvernabilité). Cependant, nous montrons ici ce résultat de manière formelle en utilisant l'algorithme de calcul de sorties plates présenté dans la section 1.3.

**Preuve.** Afin de démontrer cette proposition, nous allons appliquer les conditions nécessaires et suffisantes de platitude au modèle (3.1)-(3.6).

Commençons par construire le modèle implicite correspondant. En notant que l'équation (3.4) possède une seule entrée de commande, et en utilisant les équations (3.8) et (3.10), on obtient

$$C_{D_0}(\alpha, M) = -\frac{2m(\dot{V} + g \sin \gamma)}{\rho_0 \exp(-h/H_0) SV^2}.$$

Sous l'hypothèse que le coefficient de traînée soit à variables séparables, nous pouvons le réécrire sous la forme  $C_{D_0}(\alpha, M) = \Gamma_{C_{D_0}}(\alpha)\Lambda_{C_{D_0}}(M)$ . Nous verrons que c'est souvent le cas en pratique pour les fonctions suffisamment lisses, et nous modéliserons les coefficients aérodynamiques dans la section 3.3 en utilisant une approche découplante efficace, basée sur la décomposition en valeurs singulières de leurs matrices de définition.

De plus, si la fonction  $\Gamma_{C_{D_0}}$  est bijective sur une plage de valeurs d'incidences appropriée, alors  $\Gamma_{C_{D_0}}^{-1}(\cdot)$  existe, et nous pouvons écrire la solution  $\bar{\alpha}$  sous la forme

$$\bar{\alpha} = \Gamma_{C_{D_0}}^{-1} \left( -\frac{2m(\dot{V} + g \sin \gamma)}{\rho_0 \exp(-h/H_0) SV^2 \Lambda_{C_{D_0}}(M)} \right), \quad (3.11)$$

qui ne présente pas de singularités car  $V > 0$  et  $\Lambda_{C_{D_0}}(M) > 0$ .

**Remarque 3.2.** De même que pour le coefficient de traînée, le calcul de l'angle de dérapage  $\beta$  en fonction de la sortie plate nécessite que le coefficient de force latérale  $C_{Y_0}$  soit à variables séparables, i.e.,  $C_{Y_0}(\beta, M) = \Gamma_{C_{Y_0}}(\beta)\Lambda_{C_{Y_0}}(M)$  et que la fonction  $\Gamma_{C_{Y_0}}$  soit bijective.

Maintenant, en injectant l'expression de l'angle d'incidence  $\bar{\alpha}$  dans (3.5) et (3.6), on peut éliminer l'équation (3.4). En mettant pour le moment de côté les équations (3.1)-(3.3), on obtient le nouveau système

$$\begin{cases} \dot{\gamma} &= \frac{1}{mV} (L(\bar{\alpha}, M) \cos \mu - Y(\beta, M) \sin \mu) - \frac{g}{V} \cos \gamma, \\ \dot{\chi} &= \frac{1}{mV \cos \gamma} (L(\bar{\alpha}, M) \sin \mu + Y(\beta, M) \cos \mu), \end{cases} \quad (3.12)$$

En éliminant la force latérale  $Y(\beta, M)$  dans (3.12), on obtient finalement l'équation

$$\dot{\gamma} = \frac{L(\bar{\alpha}, M)}{mV} (\cos \mu + \tan \mu \sin \mu) - \dot{\chi} \tan \mu \cos \gamma - \frac{g}{V} \cos \gamma,$$

qui dépend uniquement de l'angle de gîte  $\mu$ . Par conséquent, le système implicite  $F(x, \dot{x})$  équivalent à (3.1)-(3.6) est donné par les trois premières équations uniquement, i.e.,

$$F(x, \dot{x}) = \begin{pmatrix} \dot{x} - V \cos \chi \cos \gamma \\ \dot{y} - V \sin \chi \cos \gamma \\ \dot{h} - V \sin \gamma \end{pmatrix} = 0. \quad (3.13)$$

En calculant la dérivée de Cartan du modèle (3.13), on obtient le système variationnel

$$P(F) = \begin{pmatrix} \frac{d}{dt} & 0 & 0 & -\cos \chi \cos \gamma & V \cos \chi \sin \gamma & V \sin \chi \cos \gamma \\ 0 & \frac{d}{dt} & 0 & -\sin \chi \cos \gamma & V \sin \chi \sin \gamma & -V \cos \chi \cos \gamma \\ 0 & 0 & \frac{d}{dt} & -\sin \gamma & -V \cos \gamma & 0 \end{pmatrix}.$$

D'après l'algorithme de platitude 1, nous commençons par calculer une forme de Smith de  $P(F)$  telle que  $VP(F)U = (I_3, O_{3,3})$ , où  $U \in \mathbf{D}\text{-Smith}(P(F))$ . En définissant successivement les matrices de transformation unimodulaires

$$U_1 = \begin{pmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}, \quad U_2 = \begin{pmatrix} -\frac{1}{\cos \chi \cos \gamma} & 0 & 0 & \frac{1}{\cos \chi \cos \gamma} \frac{d}{dt} & V \tan \gamma & V \tan \chi \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix},$$

$$U_3 = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \end{pmatrix}, \quad U_4 = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ \frac{\sin \chi}{V \cos \gamma} & -\frac{\cos \chi}{V \cos \gamma} & 0 & -\frac{\sin \chi}{V \cos \gamma} \frac{d}{dt} & 0 & \frac{\cos \chi}{V \cos \gamma} \frac{d}{dt} \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix},$$

$$U_5 = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix},$$

et

$$U_6 = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ \frac{\sin \gamma \cos \chi}{V} & \frac{\sin \chi \sin \gamma}{V} & -\frac{\cos \gamma}{V} & -\frac{\sin \gamma \cos \chi}{V} \frac{d}{dt} & \frac{\cos \gamma}{V} \frac{d}{dt} & -\frac{\sin \chi \sin \gamma}{V} \frac{d}{dt} \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix},$$

on obtient alors la matrice de transformation globale  $U = U_1U_2U_3U_4U_5U_6$  définie par

$$U = \begin{pmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ -\cos \chi \cos \gamma & -\sin \chi \cos \gamma & -\sin \gamma & \cos \chi \cos \gamma \frac{d}{dt} & \sin \gamma \frac{d}{dt} & \sin \chi \cos \gamma \frac{d}{dt} \\ \frac{\sin \gamma \cos \chi}{V} & \frac{\sin \chi \sin \gamma}{V} & -\frac{\cos \gamma}{V} & -\frac{\sin \gamma \cos \chi}{V} \frac{d}{dt} & \frac{\cos \gamma}{V} \frac{d}{dt} & -\frac{\sin \chi \sin \gamma}{V} \frac{d}{dt} \\ \frac{\sin \chi}{V \cos \gamma} & -\frac{\cos \chi}{V \cos \gamma} & 0 & -\frac{\sin \chi}{V \cos \gamma} \frac{d}{dt} & 0 & \frac{\cos \chi}{V \cos \gamma} \frac{d}{dt} \end{pmatrix},$$

telle que  $P(F)U = (I_3, 0_{3,3})$ . La matrice  $\hat{U}$  est ensuite donnée par

$$\hat{U} = U \begin{pmatrix} 0_{3,3} \\ I_3 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \\ \cos \chi \cos \gamma \frac{d}{dt} & \sin \gamma \frac{d}{dt} & \sin \chi \cos \gamma \frac{d}{dt} \\ -\frac{\sin \gamma \cos \chi}{V} \frac{d}{dt} & \frac{\cos \gamma}{V} \frac{d}{dt} & -\frac{\sin \chi \sin \gamma}{V} \frac{d}{dt} \\ -\frac{\sin \chi}{V \cos \gamma} \frac{d}{dt} & 0 & \frac{\cos \chi}{V \cos \gamma} \frac{d}{dt} \end{pmatrix}.$$

Afin de déterminer une base de l'idéal de formes différentielles  $\Omega$ , on calcule ensuite une décomposition de Smith de  $\hat{U}$  telle que  $Q\hat{U}R = (I_3, 0_{3,3})^T$ . On obtient alors une matrice de transformation unimodulaire à droite  $R$  (optionnelle) telle que

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

et les matrices de transformation à gauche définies par

$$Q_1 = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ -\cos \chi \cos \gamma \frac{d}{dt} & 0 & 0 & 1 & 0 & 0 \\ \frac{\sin \gamma \cos \chi}{V} \frac{d}{dt} & 0 & 0 & 0 & 1 & 0 \\ \frac{\sin \chi}{V \cos \gamma} \frac{d}{dt} & 0 & 0 & 0 & 0 & 1 \end{pmatrix}, \quad Q_2 = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & -\sin \chi \cos \gamma \frac{d}{dt} & 0 & 1 & 0 & 0 \\ 0 & \frac{\sin \chi \sin \gamma}{V} \frac{d}{dt} & 0 & 0 & 1 & 0 \\ 0 & -\frac{\cos \chi}{V \cos \gamma} \frac{d}{dt} & 0 & 0 & 0 & 1 \end{pmatrix},$$

et

$$Q_3 = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & -\sin \gamma \frac{d}{dt} & 1 & 0 & 0 \\ 0 & 0 & -\frac{\cos \gamma}{V} \frac{d}{dt} & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix},$$

telles que la matrice de transformation unimodulaire globale  $Q \in \mathbf{G}\text{-Smith}(\hat{U})$ , définie par

$$Q = Q_3Q_2Q_1 = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ -\cos \chi \cos \gamma \frac{d}{dt} & -\sin \chi \cos \gamma \frac{d}{dt} & -\sin \gamma \frac{d}{dt} & 1 & 0 & 0 \\ \frac{\sin \gamma \cos \chi}{V} \frac{d}{dt} & \frac{\sin \gamma \sin \chi}{V} \frac{d}{dt} & -\frac{\cos \gamma}{V} \frac{d}{dt} & 0 & 1 & 0 \\ \frac{\sin \chi}{V \cos \gamma} \frac{d}{dt} & -\frac{\cos \chi}{V \cos \gamma} \frac{d}{dt} & 0 & 0 & 0 & 1 \end{pmatrix}$$

vérifie

$$Q\hat{U}R = \begin{pmatrix} I_3 \\ 0_{3,3} \end{pmatrix}.$$

Une base  $\omega$  de l'idéal de formes différentielles  $\Omega$  est donc donnée par le vecteur de 1-formes

$$\omega = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{pmatrix} Q \begin{pmatrix} dx \\ dy \\ dh \\ dV \\ d\gamma \\ d\chi \end{pmatrix} = \begin{pmatrix} dx \\ dy \\ dh \end{pmatrix}.$$

La base  $\omega$  est trivialement fortement fermée avec  $M = I_3$ , ce qui donne finalement la sortie plate  $y_1 = x$ ,  $y_2 = y$  et  $y_3 = h$ , i.e. la position cartésienne du véhicule dans un repère local lié au sol. ■

**Remarque 3.3.** (a) Le calcul de la matrice  $R$  associée à la seconde décomposition de Smith est optionnel ici, dans le sens où il entraîne uniquement une permutation des composantes de la sortie plate.

(b) À partir de la connaissance de la sortie plate  $(x, y, h)$ , on peut bien sûr calculer les expressions des commandes  $\alpha$ ,  $\beta$  et  $\mu$  grâce aux relations (3.11) et (3.12), qui dépendent alors de la sortie plate et de ses dérivées jusqu'à l'ordre deux.

Dans la pratique, l'angle de dérapage  $\beta$  est choisi nul lors de l'optimisation d'une trajectoire de rentrée atmosphérique de référence. Plus précisément, on impose souvent dans le cas nominal une contrainte de vol symétrique, le degré de liberté associé à l'angle de dérapage permettant ensuite au pilote de compenser certaines perturbations exogènes, comme des rafales de vent par exemple, ou des variations d'efficacité de gouvernes.

Si l'on impose  $\beta = 0$  dans les équations (3.4)-(3.6), alors un observable autonome associé à l'une de ces équations va apparaître. En reprenant le système (3.12) avec la contrainte  $\beta = 0$ , alors l'élément de torsion provient de l'incapacité à satisfaire simultanément les deux équations<sup>14</sup>. Dans ce cas, le système n'est pas plat. Il est cependant possible de paramétrer l'observable autonome en fonction de la sortie plate correspondant à la partie sans torsion du système, et de vérifier que l'équation égalitaire associée est satisfaite dans l'espace de la sortie plate. Nous utiliserons cette démarche pour optimiser des trajectoires de rentrée atmosphérique dans la seconde partie de ce mémoire.

### 3.3 Platitude du modèle “Terre tournante”

Nous considérons à présent le modèle complet, à trois degrés de liberté, d'un véhicule de rentrée atmosphérique, surtout utilisé lors de la phase hypersonique, entre le point

<sup>14</sup>si l'on enlève l'une des deux équations, alors le système est plat, mais il perd son sens physique.



d'injection (après désorbitation) et le début de la phase de dissipation d'énergie en zone terminale. La principale différence avec le modèle utilisé dans la section précédente est l'apparition des forces de Coriolis (en  $\Omega V$ ) et des forces centrifuges (en  $\Omega^2 R$ ), provenant du mouvement de rotation de la Terre, ainsi que la description du modèle en fonction de coordonnées géographiques.

La non platitude du modèle en Terre non tournante, ayant pour commandes l'angle d'incidence  $\alpha$  et l'angle de gîte aérodynamique  $\mu$ , a été montrée dans [33], où les auteurs ont formalisé ce résultat et donné une paramétrisation de la partie sans torsion du système. Nous allons utiliser l'approche algébrique décrite dans la section 1.3, et l'algorithme 1, afin de déterminer, par moyen de calcul formel, une sortie plate du modèle complètement actionné, c'est à dire ayant pour commandes  $\alpha$ ,  $\beta$  et  $\mu$ . Sous l'hypothèse "Terre tournante", le modèle trois axes complet du véhicule est défini par [32, 34] (voir annexe C)

$$\dot{\varphi} = \frac{V \cos \gamma \cos \chi}{R}, \quad (3.14)$$

$$\dot{\lambda} = \frac{V \cos \gamma \sin \chi}{R \cos \varphi}, \quad (3.15)$$

$$\dot{R} = V \sin \gamma, \quad (3.16)$$

$$\dot{V} = -\frac{D(\alpha, M)}{m} - g \sin \gamma + \Omega^2 R \cos \varphi (\sin \gamma \cos \varphi - \cos \chi \cos \gamma \sin \varphi), \quad (3.17)$$

$$\begin{aligned} \dot{\gamma} &= \frac{1}{mV} (L(\alpha, M) \cos \mu - Y(\beta, M) \sin \mu) + \left( \frac{V}{R} - \frac{g}{V} \right) \cos \gamma + 2\Omega \sin \chi \cos \varphi \\ &+ \frac{\Omega^2 R}{V} \cos \varphi (\cos \chi \sin \gamma \sin \varphi + \cos \gamma \cos \varphi), \end{aligned} \quad (3.18)$$

$$\begin{aligned} \dot{\chi} &= \frac{1}{mV \cos \gamma} (L(\alpha, M) \sin \mu + Y(\beta, M) \cos \mu) + \frac{V}{R} \cos \gamma \sin \chi \tan \varphi \\ &+ 2\Omega (\sin \varphi - \cos \chi \tan \gamma \cos \varphi) + \frac{\Omega^2 R \sin \chi \sin \varphi \cos \varphi}{V \cos \gamma}, \end{aligned} \quad (3.19)$$

où  $g = \frac{\mu_T}{R^2}$  représente l'accélération gravitationnelle et  $\mu_T$  la constante gravitationnelle terrestre. La position est exprimée en coordonnées géographiques (latitude  $\varphi$ , longitude  $\lambda$ , rayon-vecteur  $R = h + R_T$ ) et la vitesse relative est exprimée dans un repère géocentrique local (module  $V$ , pente  $\gamma$ , cap  $\chi$ ). Le rayon terrestre  $R_T$  et la masse du véhicule  $m$  sont supposés être constants. Les entrées de commande sont l'incidence  $\alpha$ , le dérapage  $\beta$ , supposé être non nul pour le moment, et la gîte  $\mu$ . Les forces de portance  $L$ , de traînée  $D$  et la force latérale  $Y$  sont typiquement définies par

$$L(\alpha, M) = \bar{q} SC_{L_0}(\alpha, M), \quad (3.20)$$

$$D(\alpha, M) = \bar{q} SC_{D_0}(\alpha, M). \quad (3.21)$$

$$Y(\beta, M) = \bar{q} SC_{Y_0}(\beta, M), \quad (3.22)$$

où  $\bar{q} = \frac{1}{2}\rho V^2$  est la pression dynamique. Finalement, la densité atmosphérique  $\rho$  est une fonction décroissante de l'altitude.

Sous ces hypothèses, et avec les notations précédentes, nous obtenons la proposition suivante.

**Proposition 3.2.** *Le modèle non linéaire à trois degrés de liberté d'un véhicule de rentrée atmosphérique défini par (3.14)-(3.19) est plat, et la sortie plate est constituée des coordonnées géographiques  $(\varphi, \lambda, R)$ .*

**Preuve.** Afin de démontrer cette proposition, nous allons appliquer à nouveau les conditions nécessaires et suffisantes de platitude au modèle formé par les équations (3.14)-(3.19).

Nous utilisons la même démarche que dans la section 3.2 afin de mettre le système (3.14)-(3.19) sous forme implicite avant d'appliquer l'algorithme 1. Commençons par calculer l'expression de l'angle d'incidence  $\alpha$  à partir des équations (3.17) et (3.21). Il vient :

$$C_{D_0}(\alpha, M) = -\frac{2m}{\rho S V^2} \left( \dot{V} + g \sin \gamma - \Omega^2 R \cos \varphi (\sin \gamma \cos \varphi - \cos \chi \cos \gamma \sin \varphi) \right).$$

Sous l'hypothèse que le coefficient de traînée soit à variables séparables, nous pouvons le réécrire sous la forme  $C_{D_0}(\alpha, M) = \Gamma_{C_{D_0}}(\alpha) \Lambda_{C_{D_0}}(M)$ . De plus, si  $\Gamma_{C_{D_0}}$  est une fonction bijective, alors  $\Gamma_{C_{D_0}}^{-1}(\cdot)$  existe, et nous pouvons écrire la solution  $\bar{\alpha}$  sous la forme

$$\bar{\alpha} = \Gamma_{C_{D_0}}^{-1} \left( -\frac{2m}{\rho S V^2 \Lambda_{C_{D_0}}(M)} \left( \dot{V} + g \sin \gamma - \Omega^2 R \cos \varphi (\sin \gamma \cos \varphi - \cos \chi \cos \gamma \sin \varphi) \right) \right), \quad (3.23)$$

qui ne présente pas de singularités car  $V > 0$  et  $\Lambda_{C_{D_0}}(M) > 0$ . En utilisant l'expression  $\bar{\alpha}$ , alors le système équivalent à (3.17)-(3.19) est donné par

$$\begin{cases} \dot{\gamma} = \frac{1}{mV} (L(\bar{\alpha}, M) \cos \mu - Y(\beta, M) \sin \mu) + \left( \frac{V}{R} - \frac{g}{V} \right) \cos \gamma + 2\Omega \sin \chi \cos \varphi \\ \quad + \frac{\Omega^2 R}{V} \cos \varphi (\cos \chi \sin \gamma \sin \varphi + \cos \gamma \cos \varphi), \\ \dot{\chi} = \frac{1}{mV \cos \gamma} (L(\bar{\alpha}, M) \sin \mu + Y(\beta, M) \cos \mu) + \frac{V}{R} \cos \gamma \sin \chi \tan \varphi \\ \quad + 2\Omega (\sin \varphi - \cos \chi \tan \gamma \cos \varphi) + \frac{\Omega^2 R \sin \chi \sin \varphi \cos \varphi}{V \cos \gamma}, \end{cases} \quad (3.24)$$

Finalement, en éliminant la force latérale  $Y(\beta, M)$  dans (3.24), on obtient

$$\begin{aligned} \dot{\gamma} &= \frac{L(\bar{\alpha}, M)}{mV} (\cos \mu + \tan \mu \sin \mu) - \dot{\chi} \tan \mu \cos \gamma - \frac{g}{V} \cos \gamma \\ &+ \frac{V}{R} \cos \gamma (1 + \cos \gamma \tan \mu \sin \chi \tan \varphi) \\ &+ 2\Omega (\sin \chi \cos \varphi - \cos \gamma \tan \mu (\sin \varphi - \cos \chi \tan \gamma \cos \varphi)) \\ &+ \frac{\Omega^2 R}{V} \cos \varphi (\cos \chi \sin \gamma \sin \varphi + \cos \gamma \cos \varphi - \tan \mu \sin \chi \sin \varphi), \end{aligned} \quad (3.25)$$

qui dépend uniquement de l'angle de gîte  $\mu$ . Le système implicite  $F(x, \dot{x})$  équivalent à (3.14)-(3.19) est alors donné par

$$F(x, \dot{x}) = \begin{pmatrix} \dot{\varphi} - \frac{V \cos \gamma \cos \chi}{R} \\ \dot{\lambda} - \frac{V \cos \gamma \sin \chi}{R \cos \varphi} \\ \dot{R} - V \sin \gamma \end{pmatrix} = 0. \quad (3.26)$$

Le système variationnel  $P(F)$  associé est alors donné par

$$P(F) = \begin{pmatrix} \frac{d}{dt} & 0 & 0 & -\sin \gamma & -V \cos \gamma & 0 \\ \frac{V \cos \chi \cos \gamma}{R^2} & \frac{d}{dt} & 0 & -\frac{\cos \gamma \cos \chi}{R} & \frac{V \sin \gamma \cos \chi}{R} & \frac{V \cos \gamma \sin \chi}{R} \\ \frac{V \cos \gamma \sin \chi}{R^2 \cos \varphi} & -\frac{V \cos \gamma \sin \chi \sin \varphi}{R \cos^2 \varphi} & \frac{d}{dt} & -\frac{\sin \chi \cos \gamma}{R \cos \varphi} & \frac{V \sin \chi \sin \gamma}{R \cos \varphi} & -\frac{V \cos \chi \cos \gamma}{R \cos \varphi} \end{pmatrix}.$$

Commençons par calculer une matrice de transformation unimodulaire  $U \in \text{D-Smith}(P(F))$  telle que  $VP(F)U = (I_3, O_{3,3})$ . En utilisant l'algorithme 13, on définit successivement les matrices élémentaires suivantes :

$$U_1 = \begin{pmatrix} 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}, \quad U_2 = \begin{pmatrix} -\frac{1}{V \cos \gamma} & 0 & 0 & -\frac{\tan \gamma}{V} & \frac{1}{V \cos \gamma} \frac{d}{dt} & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix},$$

$$U_3 = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix},$$

$$U_4 = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ -\sin \gamma & -\frac{R \cos \gamma}{\cos \chi} & 0 & \frac{R \cos \gamma}{\cos \chi} \frac{d}{dt} & \sin \gamma \frac{d}{dt} + \frac{V \cos^2 \gamma}{R} & V \cos^2 \gamma \tan \chi \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix},$$

$$U_5 = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{pmatrix},$$

et

$$U_6 = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{R \sin \chi}{V \cos \gamma} & -\frac{R \cos \varphi \cos \chi}{V \cos \gamma} & -\frac{R \sin \chi}{V \cos \gamma} \frac{d}{dt} - \sin \chi \cos \chi \tan \varphi & 0 & \frac{R \cos \chi \cos \varphi}{V \cos \gamma} \frac{d}{dt} \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}.$$

On obtient alors la matrice de transformation globale  $U = U_1 U_2 U_3 U_4 U_5 U_6$  définie par

$$U = \begin{pmatrix} 0 & 0 & 0 & \cdots & 0 & 0 & 0 \\ 0 & 0 & 0 & \cdots & 0 & 0 & 0 \\ 0 & 0 & 0 & \cdots & 0 & 0 & 0 \\ -\sin \gamma & -R \cos \gamma \cos \chi & -R \cos \gamma \sin \chi \cos \varphi & \cdots & 0 & 0 & 0 \\ -\frac{\cos \gamma}{V} & \frac{R \sin \gamma \cos \chi}{V} & \frac{R \sin \gamma \sin \chi \cos \varphi}{V} & \cdots & 1 & 0 & 0 \\ 0 & \frac{R \sin \chi}{V \cos \gamma} & -\frac{R \cos \chi \cos \varphi}{V \cos \gamma} & \cdots & 0 & 0 & 1 \\ & 0 & 0 & & 0 & 1 & 0 \\ & 1 & 0 & & 0 & 0 & 0 \\ & 0 & 0 & & 0 & 0 & 1 \\ R \cos \gamma \cos \chi \frac{d}{dt} - V \cos^2 \gamma \tan \varphi (1 - \cos^2 \chi) & \sin \gamma \frac{d}{dt} + \frac{V \cos^2 \gamma}{R} & R \cos \gamma \sin \chi \cos \varphi \frac{d}{dt} & & & & \\ -\frac{R \sin \gamma \cos \chi}{V} \frac{d}{dt} + \sin \gamma \cos \gamma \tan \varphi (1 - \cos^2 \chi) & \frac{\cos \gamma}{V} \frac{d}{dt} - \frac{\cos \gamma \sin \gamma}{R} & -\frac{R \sin \gamma \sin \chi \cos \varphi}{V} \frac{d}{dt} & & & & \\ -\frac{R \sin \chi}{V \cos \gamma} \frac{d}{dt} - \sin \chi \cos \chi \tan \varphi & 0 & \frac{R \cos \chi \cos \varphi}{V \cos \gamma} \frac{d}{dt} & & & & \end{pmatrix},$$

vérifiant  $P(F)U = (I_3, 0_{3,3})$ . La matrice  $\hat{U}$  est ensuite donnée par

$$\hat{U} = \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \\ R \cos \gamma \cos \chi \frac{d}{dt} - V \cos^2 \gamma \tan \varphi (1 - \cos^2 \chi) & \sin \gamma \frac{d}{dt} + \frac{V \cos^2 \gamma}{R} & R \cos \gamma \sin \chi \cos \varphi \frac{d}{dt} \\ -\frac{R \sin \gamma \cos \chi}{V} \frac{d}{dt} + \sin \gamma \cos \gamma \tan \varphi (1 - \cos^2 \chi) & \frac{\cos \gamma}{V} \frac{d}{dt} - \frac{\cos \gamma \sin \gamma}{R} & -\frac{R \sin \gamma \sin \chi \cos \varphi}{V} \frac{d}{dt} \\ -\frac{R \sin \chi}{V \cos \gamma} \frac{d}{dt} - \sin \chi \cos \chi \tan \varphi & 0 & \frac{R \cos \chi \cos \varphi}{V \cos \gamma} \frac{d}{dt} \end{pmatrix}.$$

On calcule ensuite une matrice de transformation  $Q \in \mathbf{G}\text{-Smith}(\hat{U})$  telle que  $Q\hat{U}R = (I_3, 0_{3,3})^T$ . On obtient alors une matrice de transformation unimodulaire à droite  $R$  (optionnelle) telle que

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

et les matrices de transformation à gauche  $Q_1$ ,  $Q_2$  et  $Q_3$  définies par

$$Q_1 = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ -\sin \gamma \frac{d}{dt} - \frac{V \cos^2 \gamma}{R} & 0 & 0 & 1 & 0 & 0 & 0 \\ -\frac{\cos \gamma}{V} \frac{d}{dt} + \frac{\cos \gamma \sin \gamma}{R} & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix},$$

$$Q_2 = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & -R \cos \gamma \cos \chi \frac{d}{dt} + V \cos^2 \gamma \tan \varphi (1 - \cos^2 \chi) & 0 & 0 & 1 & 0 & 0 \\ 0 & \frac{R \sin \gamma \cos \chi}{V} \frac{d}{dt} - \sin \gamma \cos \gamma \tan \varphi (1 - \cos^2 \chi) & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -\frac{R \cos \chi \cos \varphi}{V \cos \gamma} \frac{d}{dt} & 0 & 0 & 1 \end{pmatrix},$$

et

$$Q_3 = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & -R \cos \gamma \sin \chi \cos \varphi \frac{d}{dt} & 1 & 0 & 0 & 0 \\ 0 & 0 & \frac{R \sin \gamma \sin \chi \cos \varphi}{V} \frac{d}{dt} & 0 & 1 & 0 & 0 \\ 0 & 0 & -\frac{R \cos \chi \cos \varphi}{V \cos \gamma} \frac{d}{dt} & 0 & 0 & 1 & 0 \end{pmatrix},$$

telles que la matrice de transformation unimodulaire globale  $Q \in \mathbf{G}\text{-Smith}(\hat{U})$ , définie par

$$Q = Q_3 Q_2 Q_1 = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & \dots \\ 0 & 1 & 0 & 0 & 0 & 0 & \dots \\ 0 & 0 & 0 & 1 & 0 & 0 & \dots \\ -\sin \gamma \frac{d}{dt} - \frac{V \cos^2 \gamma}{R} & -R \cos \gamma \cos \chi \frac{d}{dt} + V \cos^2 \gamma \tan \varphi (1 - \cos^2 \chi) & 0 & 0 & 0 & 0 & \dots \\ -\frac{\cos \gamma}{V} \frac{d}{dt} + \frac{\cos \gamma \sin \gamma}{R} & \frac{R \sin \gamma \cos \chi}{V} \frac{d}{dt} - \sin \gamma \cos \gamma \tan \varphi (1 - \cos^2 \chi) & 0 & 0 & 0 & 0 & \dots \\ 0 & \frac{R \sin \chi}{V \cos \gamma} \frac{d}{dt} + \sin \chi \cos \chi \tan \varphi & 0 & 0 & 0 & 0 & \dots \\ \dots & 0 & 0 & 0 & 0 & 0 & \dots \\ \dots & 0 & 0 & 0 & 0 & 0 & \dots \\ \dots & 1 & 0 & 0 & 0 & 0 & \dots \\ \dots & -R \cos \gamma \sin \chi \cos \varphi \frac{d}{dt} & 1 & 0 & 0 & 0 & \dots \\ \dots & \frac{R \sin \gamma \sin \chi \cos \varphi}{V} \frac{d}{dt} & 0 & 1 & 0 & 0 & \dots \\ \dots & -\frac{R \cos \chi \cos \varphi}{V \cos \gamma} \frac{d}{dt} & 0 & 0 & 1 & 0 & \dots \end{pmatrix},$$

fournit la décomposition de Smith de  $\hat{U}$ , telle que

$$Q \hat{U} R = \begin{pmatrix} I_3 \\ 0_{3,3} \end{pmatrix}.$$

Une base  $\omega$  de l'idéal de formes différentielles  $\Omega$  associé à  $P(F)$  est donc donnée par le vecteur de 1-formes

$$\omega = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{pmatrix} Q \begin{pmatrix} d\varphi \\ d\lambda \\ dR \\ dV \\ d\gamma \\ d\chi \end{pmatrix} = \begin{pmatrix} d\varphi \\ d\lambda \\ dR \end{pmatrix}.$$

La base  $\omega$  est trivialement fortement fermée avec  $M = I_3$ , ce qui donne finalement la sortie plate  $y_1 = \varphi$ ,  $y_2 = \lambda$  et  $y_3 = R$ , i.e. la position géographique du véhicule. ■

**Remarque 3.4.** Le modèle du véhicule en “Terre plate” et le modèle à trois degrés de liberté en terre tournante sont équivalents dans le sens où il existe une transformation ponctuelle qui associe à toute trajectoire de la sortie plate  $(x, y, h)$  une trajectoire de la sortie plate  $(\varphi, \lambda, R)$ , telle que

$$\begin{cases} \dot{x} = R\dot{\varphi}, \\ \dot{y} = R\dot{\lambda} \cos \varphi, \\ \dot{h} = \dot{R}. \end{cases}$$

Cette transformation ne remet bien sûr pas en cause la propriété de platitude du modèle à trois degrés de liberté obtenue dans la section 3.2.

### 3.4 Conclusion

Dans ce chapitre, nous avons montré que le modèle à trois degrés de liberté d’un véhicule de rentrée atmosphérique est plat. De plus, nous avons déterminé une sortie plate candidate associée en utilisant un algorithme de calcul formel. Deux modélisations ont été considérées, l’une étant définie sous l’hypothèse “Terre plate”, l’autre tenant compte du mouvement de rotation terrestre. Les deux formulations ont conduit à la propriété de platitude du modèle, dont une sortie plate est donnée par la position du centre de gravité du véhicule dans un repère local lié à la piste. La propriété de platitude du modèle à trois degrés de liberté du véhicule va nous permettre d’établir, dans la seconde partie de ce mémoire, une méthodologie complète pour le développement d’une loi de guidage autonome. Dans un second temps, cette loi de guidage autonome sera appliquée aux phases TAEM et A&L d’une mission de rentrée atmosphérique de la navette spatiale américaine STS-1.



## Deuxième partie

# Planification autonome de trajectoires de rentrée atmosphérique





# Position du problème

L'un des objectifs majeurs pour la conception des véhicules de rentrée atmosphérique de nouvelle génération réside dans l'amélioration de la sûreté de fonctionnement et la réduction des coûts associés. En conditions nominales, les technologies de guidage et de pilotage (G&C) actuelles sont suffisamment matures pour garantir la mise en orbite d'un véhicule et son retour sur Terre, avec un atterrissage fiable sur une piste d'atterrissage. Ce type de technologie a été éprouvé à maintes reprises lors du programme américain Shuttle. Cependant, c'est en conditions défaillantes que l'on est souvent amené à considérer des stratégies différentes, communément appelées solutions de guidage et de pilotage avancées (AG&C). En permettant aux véhicules de rentrée défaillants de regagner la Terre de manière sûre lorsque cela est physiquement possible, les solutions AG&C sont essentielles pour l'amélioration de la fiabilité et des coûts des véhicules de nouvelle génération. De plus, un autre objectif de ces techniques est de limiter la quantité d'analyses effectuées au sol pour chaque scénario défaillant considéré.

Communément, un système de guidage, navigation et pilotage (GNC) *autonome* comporte trois parties principales :

1. un module de détection et localisation de défauts (FDI),
2. des mécanismes d'adaptation des boucles de régulation interne (pilotage) et externe (guidage),
3. un module de replanification de trajectoires,

Les technologies AG&C actuelles sont capables de reconfigurer en ligne la loi de pilotage, et d'adapter la boucle de régulation du guidage. Dans la plupart des cas, les surfaces de contrôle aérodynamique disponibles sont allouées afin de compenser un ou plusieurs défauts, et de retrouver autant que possible le niveau de robustesse/performance nominal de la boucle de régulation. Ce type d'approche est souvent appelé reconfiguration de la boucle interne. Lorsque la boucle de pilotage reconfigurée ne parvient pas à regagner le niveau de robustesse/performance nominal, la loi de guidage doit être adaptée afin que la boucle interne ne conduise pas à l'instabilité du système.

Cependant, dans certaines situations, la reconfiguration de la loi de pilotage et l'adaptation de la boucle de guidage ne sont pas suffisantes, notamment lors de l'apparition de défaillances de forte amplitude ou de défauts multiples, qui peuvent réduire de manière importante l'énergie disponible pour le contrôle des forces aérodynamiques, conduisant à

une dégradation sévère des performances et à une réduction de l'enveloppe de vol. Ainsi, certaines défaillances peuvent entraîner une modification importante des forces et des moments aérodynamiques agissant sur le véhicule en configuration nominale. Il est alors nécessaire de replanifier la trajectoire de vol courante afin de générer de nouvelles consignes de référence de la boucle de guidage, se trouvant à l'intérieur de l'enveloppe de vol réduite.

Afin d'améliorer la tolérance aux défauts, plusieurs travaux de recherche se sont focalisés sur la replanification de trajectoires à bord, ainsi que l'adaptation de la boucle de guidage [35, 36, 37, 38]. La replanification de trajectoires de rentrée atmosphérique doit répondre aux contraintes suivantes :

- (a) la planification/replanification de trajectoires à bord nécessite que les algorithmes associés soient certifiables pour le vol,
- (b) l'effet des défaillances sur la dynamique du véhicule doit être prédit sur une large plage de conditions de vol.

A l'heure actuelle, la replanification de trajectoires nécessite une quantité importante d'analyses au sol. La planification de trajectoires en conditions défaillantes nécessite la génération d'une importante base de données de trajectoires admissibles pour un sous-ensemble de défaillances potentielles. Ainsi, la durée séparant le développement de l'algorithme de planification et le lancement de la mission peut s'étendre jusqu'à plusieurs mois, engendrant ainsi des délais d'exploitation importants. La réduction de ces délais et des coûts associés représente donc un problème important qui est abordé dans ce mémoire.

La contribution principale de cette partie est le développement d'une *loi de guidage autonome* permettant la replanification de trajectoires à bord en présence de situations défaillantes. L'architecture de la loi de guidage proposée comprend un *planificateur de trajectoires* tolérant aux défauts ainsi qu'un *régulateur de poursuite de trajectoires* de type linéaire à paramètres variants (LPV). L'algorithme de replanification de trajectoires est basé sur les techniques de transcription directes et la résolution d'un problème de programmation non linéaire (NLP), tenant compte de la propriété de platitude du modèle à trois degrés de liberté du véhicule. En outre, des contraintes de trajectoire additionnelles, résultant des défaillances simples ou multiples de gouvernes, sont incluses dans le problème de commande optimale afin que le véhicule dégradé puisse être équilibré statiquement le long de la trajectoire replanifiée. En outre, une méthodologie originale d'annexion super-ellipsoïdale est proposée afin de transformer le problème de commande optimale, formulé dans l'espace de la sortie plate, en un problème convexe. L'ensemble de la méthodologie de guidage autonome est ensuite appliqué à une mission de rentrée atmosphérique dans le chapitre 3, constituée des phases de dissipation d'énergie en zone terminale (TAEM), et d'approche et d'atterrissage automatique (A&L), pour la navette spatiale américaine STS-1.

# Chapitre 1

## État de l'art

“Jamais les mots ne manquent aux idées ; ce sont les idées qui manquent aux mots.”

**Joseph Joubert**

---

### Résumé :

*Ce chapitre est consacré à la présentation des principales techniques de programmation non linéaire (NLP) et de planification de trajectoires.*

*Après avoir décrit la méthode classique de Newton, nous présenterons le cadre de programmation quadratique séquentielle (SQP), principalement dédié à la résolution des problèmes de type NLP.*

*Dans un second temps, nous analyserons les principales techniques d'optimisation de trajectoires, telles que les méthodes de tirs simples/multiples, ainsi que les méthodes basées sur une transcription directe ou indirecte du problème de commande optimale. Chaque technique abordée sera resituée dans le contexte de l'optimisation de trajectoires de rentrée atmosphérique.*

---

## 1.1 Introduction

Dans ce premier chapitre, nous proposons de faire un tour d'horizon des principales techniques d'optimisation de trajectoires rencontrées dans la littérature, et dans le paysage industriel mondial. Devant l'abondance des techniques d'optimisation de trajectoires qui ont été développées depuis une cinquantaine d'années, souvent en marge des grands programmes d'exploration spatiaux, il ne serait pas raisonnable ici de prétendre pouvoir présenter ces méthodes de manière exhaustive. Ainsi, nous décrirons deux grands courants qui ont émergé de la communauté de l'optimisation, et qui ont donné naissance à des méthodes d'optimisation complémentaires, mais au but final identique :

- les méthodes indirectes, reposant sur une résolution le plus souvent explicite des équations associées au principe du maximum de Pontryagin,
- les méthodes directes, utilisant les variables du système dynamique comme variables de décision du problème d'optimisation.

Nous nous sommes largement inspirés de l'ouvrage de J. T. Betts [34] afin de mener cette étude bibliographique.

La suite de ce chapitre s'articule de la façon suivante. Nous décrirons dans un premier temps les algorithmes numériques d'optimisation, tels que les techniques de Programmation Quadratique (QP) et de Programmation Non Linéaire (NLP), permettant de résoudre un problème d'optimisation de dimension finie. Dans un second temps, nous nous attacherons à montrer comment un problème d'optimisation de trajectoires peut être mis sous la forme d'un problème NLP, en utilisant des techniques de tirs simples ou multiples, ou des méthodes de transcription directe ou indirecte du problème de commande optimale (OCP).

## 1.2 Programmation non linéaire

### 1.2.1 Méthode de Newton

La méthode de Newton [39], introduite il y a plus de 300 ans, est sans doute la technique itérative la plus répandue pour résoudre des problèmes d'optimisation. Nous verrons que sa généralisation est au coeur d'un des algorithmes de programmation non linéaire les plus utilisés, c'est à dire la méthode de *programmation quadratique séquentielle* (SQP) [34].

Commençons par décrire la méthode de Newton à plusieurs variables. Considérons pour cela un vecteur  $x$  de dimension  $n$ , tel que  $x = (x_1, \dots, x_n)^T$ , et cherchons à vérifier un ensemble d'équations algébriques non linéaires défini par

$$c(x) = \begin{pmatrix} c_1(x) \\ \vdots \\ c_n(x) \end{pmatrix} = 0. \quad (1.1)$$

En développant l'expression (1.1) en série de Taylor au premier ordre, on obtient

$$c(\bar{x}) = c(x) + G(\bar{x} - x), \quad (1.2)$$

où la matrice jacobienne  $G$  est définie par

$$G \equiv \frac{\partial c}{\partial x} = \begin{pmatrix} \frac{\partial c_1}{\partial x_1} & \frac{\partial c_1}{\partial x_2} & \dots & \frac{\partial c_1}{\partial x_n} \\ \frac{\partial c_2}{\partial x_1} & \frac{\partial c_2}{\partial x_2} & \dots & \frac{\partial c_2}{\partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial c_m}{\partial x_1} & \frac{\partial c_m}{\partial x_2} & \dots & \frac{\partial c_m}{\partial x_n} \end{pmatrix}. \quad (1.3)$$

Clairement, comme nous cherchons à obtenir  $c(\bar{x}) = 0$ , l'équation (1.2) se réécrit sous forme matricielle

$$Gp = -c. \quad (1.4)$$

Si l'on suppose que la matrice jacobienne  $G$  est non singulière, la résolution du système d'équations linéaires (1.4) fournit alors la *direction de recherche*  $p$ , qui conduit à une itération de la forme

$$\bar{x} = x + p.$$

### 1.2.2 Problème d'optimisation sans contraintes

Considérons à présent un problème d'optimisation multidimensionnel sans contraintes. Nous souhaitons trouver un vecteur de variables  $x = (x_1, \dots, x_n)^T$  de dimension  $n$ , permettant de minimiser la fonction  $F(x) = F(x_1, \dots, x_n)$ . A l'instar de la section précédente, calculons une approximation au second ordre de  $F(x)$  autour du point  $x$ , telle que

$$F(\bar{x}) = F(x) + g^T(x)(\bar{x} - x) + \frac{1}{2}(\bar{x} - x)^T H(x)(\bar{x} - x), \quad (1.5)$$

où  $g(x)$  représente le vecteur de gradients de dimension  $n$  défini par

$$g(x) \equiv \nabla_x F = \begin{pmatrix} \frac{\partial F}{\partial x_1} \\ \vdots \\ \frac{\partial F}{\partial x_n} \end{pmatrix},$$

et  $H(x)$  est la matrice hessienne de taille  $n \times n$  telle que

$$H(x) \equiv \nabla_{xx}^2 F = \begin{pmatrix} \frac{\partial^2 F}{\partial x_1^2} & \frac{\partial^2 F}{\partial x_1 \partial x_2} & \dots & \frac{\partial^2 F}{\partial x_1 \partial x_n} \\ \frac{\partial^2 F}{\partial x_2 \partial x_1} & \frac{\partial^2 F}{\partial x_2^2} & \dots & \frac{\partial^2 F}{\partial x_2 \partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 F}{\partial x_n \partial x_1} & \frac{\partial^2 F}{\partial x_n \partial x_2} & \dots & \frac{\partial^2 F}{\partial x_n^2} \end{pmatrix}. \quad (1.6)$$

En notant  $p = \bar{x} - x$  la direction de recherche, l'expression (1.5) peut se réécrire

$$F(\bar{x}) = F(x) + g^T p + \frac{1}{2} p^T H p. \quad (1.7)$$

Le terme scalaire  $g^T p$  est communément appelé *dérivée directionnelle* le long de  $p$ , et le scalaire  $p^T H p$  est appelé *courbure* ou *dérivée directionnelle du deuxième ordre* dans la direction  $p$ .

Supposons maintenant que la série (1.7) a été développée autour du point minimal  $x^*$ . Si  $x^*$  est un minimum local, alors la fonction objectif doit être supérieure pour tout point dans son voisinage, i.e.,  $F(\bar{x}) > F(x^*)$ . Cela se traduit par le fait que les gradients dans toutes les directions doivent être nuls, i.e.,  $(g^*)^T p = 0$ , ou encore

$$g(x^*) = \begin{pmatrix} g_1(x^*) \\ \vdots \\ g_n(x^*) \end{pmatrix} = 0.$$

De plus, si la fonction s'incurve dans toutes les directions, alors  $x^*$  est un minimum local fort, et le troisième terme du développement (1.7) doit être positif ou nul, i.e.,

$$p^T H^* p \geq 0. \quad (1.8)$$

Par conséquent,  $H$  est une matrice (semi) définie positive, ce qui fournit les conditions locales d'optimalité suivantes :

*Conditions nécessaires*

$$\begin{aligned} g(x^*) &= 0, \\ p^T H^* p &\geq 0. \end{aligned}$$

*Conditions suffisantes*

$$\begin{aligned} g(x^*) &= 0, \\ p^T H^* p &> 0. \end{aligned}$$

Considérons maintenant le développement quadratique (1.7) autour d'un point  $x$  arbitraire. Nous souhaitons obtenir un nouveau point  $\bar{x}$  tel que le gradient soit nul. Son approximation linéaire est alors simplement donnée par

$$\bar{g} = 0 = g + H p. \quad (1.9)$$

La résolution du système linéaire (1.9) fournit alors la direction de recherche  $p$  telle que

$$p = -H^{-1} g, \quad (1.10)$$

et l'itération de Newton est alors donnée par  $\bar{x} = x + p$ . De plus, afin de garantir la convergence de cette itération vers un minimum (local ou global), il est nécessaire de rajouter une contrainte permettant d'assurer la "descente" du gradient, telle que

$$g^T p < 0. \quad (1.11)$$

**Remarque 1.1.** (a) En utilisant la direction de recherche (1.10), la contrainte (1.11) sur le gradient s'écrit

$$g^T p = -g^T H^{-1} g < 0,$$

qui est vérifiée si seulement si la matrice hessienne est définie positive, i.e., (1.8) est satisfaite.

- (b) La question de l'inversibilité de la matrice hessienne  $H$  ne se pose pas ici pour la résolution du système (1.10) car toute matrice (semi) définie positive est inversible, et son inverse est également une matrice (semi) définie positive.

Comme nous l'avons vu précédemment, la méthode d'optimisation de Newton nécessite le calcul d'informations dérivées, telles que la matrice jacobienne (1.3) ou la matrice hessienne (1.6). Ces calculs sont particulièrement coûteux, et peuvent être approximés de manière itérative en utilisant les méthodes de type quasi-Newton. Ainsi, l'idée de base est de construire un nouvel estimé des matrices jacobienne et hessienne à partir des informations contenues dans les itérations précédentes, tel que

$$\bar{B} = B + \mathcal{R}(\Delta c, \Delta x), \quad (1.12)$$

où le nouvel estimé  $\bar{B}$  est calculé à partir de l'estimé  $B$  précédent, et

$$\begin{aligned} \Delta c &= c^k - c^{k-1}, \\ \Delta x &= x^k - x^{k-1}. \end{aligned}$$

Parmi les méthodes les plus courantes, permettant de rafraîchir la matrice  $B$  sous la forme (1.12), on peut distinguer :

- la *mise à jour de Broyden* ou *sécante*, de rang 1, telle que

$$\bar{B} = B + \frac{(\Delta c - B\Delta x)(\Delta x)^T}{(\Delta x)^T(\Delta x)},$$

- la *mise à jour symétrique de rang 1* (SR1), définie par

$$\bar{B} = B + \frac{(\Delta g - B\Delta x)(\Delta g - B\Delta x)^T}{(\Delta g - B\Delta x)^T \Delta x},$$

où  $\Delta g \equiv g^k - g^{k-1}$ .

- la *mise à jour BFGS*, sécante définie positive de rang 2, telle que

$$\bar{B} = B + \frac{\Delta g(\Delta g)^T}{(\Delta g)^T \Delta x} - \frac{B\Delta x(\Delta x)^T B}{(\Delta x)^T B \Delta x}.$$

- la *mise à jour DFP*, sécante définie positive inverse, définie par

$$\bar{B} = B + \frac{(\Delta g - B\Delta x)(\Delta g)^T + \Delta g(\Delta g - B\Delta x)^T}{(\Delta g)^T \Delta x} - \frac{(\Delta g - B\Delta x)^T \Delta x}{((\Delta g)^T \Delta x)^2} \Delta g(\Delta g)^T.$$

### 1.2.3 optimisation sous contraintes égalitaires

Dans les sections précédentes, nous avons vu comment satisfaire des contraintes égalitaires du type  $c(x) = 0$ , et comment trouver le minimum d'une fonction  $F(x)$  à partir de la méthode de Newton. Nous cherchons à présent à réaliser les deux optimisations ensemble, c'est à dire déterminer un ensemble de variables  $x = (x_1, \dots, x_n)$  minimisant la fonction  $F(x)$ , sous les  $m$  contraintes égalitaires

$$c(x) = 0,$$



avec  $m \leq n$ .

L'approche classique pour résoudre ce type de problème consiste à introduire le *lagrangien*

$$L(x, \lambda) = F(x) - \lambda^T c(x) = F(x) - \sum_{i=1}^m \lambda_i c_i(x), \quad (1.13)$$

où  $\lambda$  représente un vecteur de *multiplieurs de Lagrange*, de dimension  $m$ . Les conditions d'optimalité, exprimées pour la fonction lagrangienne (1.13), nécessitent à la fois que les dérivées de cette dernière par rapport à  $x$  et  $\lambda$  soient nulles, i.e.,  $(x^*, \lambda^*)$  est un optimum si

$$\nabla_x L(x^*, \lambda^*) = 0, \quad (1.14)$$

$$\nabla_\lambda L(x^*, \lambda^*) = 0. \quad (1.15)$$

$$(1.16)$$

Le gradient de  $L$  par rapport à  $x$  s'écrit

$$\nabla_x L = g - G^T \lambda = \nabla_F - \sum_{i=1}^m \lambda_i \nabla c_i, \quad (1.17)$$

et le gradient de  $L$  par rapport à  $\lambda$  est défini par

$$\nabla_\lambda L = -c(x). \quad (1.18)$$

Comme pour l'optimisation sans contraintes, il est nécessaire d'imposer une contrainte supplémentaire liée à la direction de descente du gradient du lagrangien afin d'assurer la convergence vers un minimum local ou global. Pour cela, on définit le *hessien du lagrangien* sous la forme

$$H_L = \nabla_{xx}^2 L = \nabla_{xx}^2 F - \sum_{i=1}^m \lambda_i \nabla_{xx}^2 c_i.$$

Il s'en suit qu'une condition suffisante de descente est donnée par

$$v^T H_L v > 0,$$

pour tout vecteur  $v$  dans l'espace tangent des contraintes.

Nous sommes maintenant en mesure d'appliquer la méthode de Newton au problème d'optimisation sous contraintes égalitaires, de manière à déterminer les valeurs de  $(x, \lambda)$  telles que les conditions (1.14)-(1.15) soient satisfaites. Commençons par calculer, de manière analogue à (1.2), un développement en série de Taylor des équations (1.17) et (1.18) au premier ordre :

$$0 = g - G^T \lambda + H_L(\bar{x} - x) - G^T(\bar{\lambda} - \lambda), \quad (1.19)$$

$$0 = -c - G(\bar{x} - x). \quad (1.20)$$

Après simplification de ces équations, on obtient un système linéaire similaire à (1.4), appelé système de Kuhn-Tucker (KT) ou de Karush-Kuhn-Tucker (KKT) [40], défini par

$$\begin{pmatrix} H_L & G^T \\ G & 0 \end{pmatrix} \begin{pmatrix} -p \\ -\bar{\lambda} \end{pmatrix} = \begin{pmatrix} g \\ c \end{pmatrix}, \quad (1.21)$$

où  $p$  représente la direction de recherche pour une itération  $\bar{x} = x + p$ , et  $\bar{\lambda}$  est le vecteur des multiplicateurs de Lagrange au nouveau point  $(\bar{x}, \bar{\lambda})$ .

Il est possible de montrer que le système (1.21) représente les conditions d'optimalité d'un sous-problème d'optimisation quadratique linéaire, défini par

$$g^T p + \frac{1}{2} p^T H p, \quad (1.22)$$

sous les contraintes

$$Gp = -c. \quad (1.23)$$

#### 1.2.4 Optimisation sous contraintes inégalitaires

Nous cherchons à présent à déterminer un vecteur  $x = (x_1, \dots, x_n)$  permettant de minimiser

$$F(x),$$

sous les  $m$  contraintes inégalitaires

$$c(x) \geq 0, \quad (1.24)$$

où  $m \geq n$ . Un point satisfaisant les contraintes (1.24) est dit *admissible*, et l'ensemble de tous les points admissibles s'appelle la *région admissible*. Les problèmes d'optimisation sous contraintes inégalitaires sont caractérisés par un concept fondamental, celui d'ensemble actif de contraintes. Ainsi, à la solution optimale  $x^*$ ,

- certaines contraintes vont être satisfaites en tant que contraintes égalitaires, i.e.,

$$c_i(x^*) = 0, \quad \forall i \in \mathcal{A},$$

où  $\mathcal{A}$  est appelé l'*ensemble actif*,

- d'autres contraintes seront satisfaites de manière stricte, i.e.,

$$c_i(x^*) > 0, \quad \forall i \in \mathcal{A}',$$

où  $\mathcal{A}'$  est appelé l'*ensemble inactif*.

Les contraintes sont donc séparées en deux sous-catégories distinctes : les contraintes actives et les contraintes inactives. Clairement, les premières (contraintes égalitaires) peuvent être traitées par la méthode de Newton décrite dans la section précédente. Les secondes nécessitent un mécanisme permettant de classifier les différentes contraintes, appelé *stratégie d'ensemble actif* et décrit dans la section suivante.

## 1.2.5 Programmation non linéaire

### Programmation quadratique

Le problème d'optimisation quadratique (1.22) fait partie intégrante d'une classe plus générale de problèmes d'optimisation, connus sous le nom de problèmes de *programmation quadratique* (QP) [34]. Un problème d'optimisation QP est caractérisé par une fonction objectif quadratique, du type

$$F(x) = g^T x + \frac{1}{2} x^T H x,$$

et des contraintes linéaires définies par

$$Ax = a, \tag{1.25}$$

$$Bx \geq b, \tag{1.26}$$

où  $H$  est la matrice hessienne définie positive.

Les grandes lignes de la méthode permettant de résoudre le problème d'optimisation QP précédent sous la forme d'une stratégie d'ensemble actif sont décrites dans l'algorithme 3.

Les *stratégies de globalisation* permettent de détecter, puis de corriger certaines déficiences associées à la méthode de Newton. Si tout se passe bien, c'est à dire si l'algorithme d'optimisation ne rencontre pas de difficulté particulière, alors les stratégies de globalisation doivent devenir transparentes au niveau global. Il existe principalement quatre techniques de globalisation largement répandues :

1. L'utilisation de *fonctions de mérite*, permettant de quantifier la progression de la convergence de l'algorithme de Newton. En effet, chaque itération de l'algorithme est de la forme  $\bar{x} = x + p$ , où la direction de recherche est, rappelons-le, obtenue en résolvant le système linéaire  $Gp = -c$ . La fonction de mérite permet alors de s'assurer que la séquence d'itérés  $\{x^{(k)}\}$  converge bien vers la solution  $x^*$ , en vérifiant que

$$M(x^{(k+1)}) < M(x^{(k)}).$$

Lorsque la méthode de Newton est utilisée dans le cadre de l'optimisation sans contraintes d'une seule fonction, alors on peut simplement prendre  $M(x) = F(x)$ . Dans le cas de fonctions multiples, une seule valeur scalaire n'est pas suffisante pour quantifier la progression de l'algorithme. La fonction de mérite la plus utilisée pour les équations non linéaires est alors définie par

$$M(x) = \frac{1}{2} c^T(x) c(x), \tag{1.27}$$

mais les normes 1, 2 et  $\infty$  du vecteur de contraintes  $c(x)$  peuvent également être utilisées de manière équivalente afin de quantifier la progression de la séquence d'itérés.

**Algorithme 3** : Algorithme de programmation quadratique (QP).**Entrées** : Un estimé de l'ensemble actif initial  $\mathcal{A}^0$  et un point initial admissible  $x^0$ .**Sorties** : La solution optimale  $(x^*, \lambda^*, \eta^*)$  du problème QP.**Algorithme** :

- 1 Calculer le minimum de la fonction objectif quadratique, sujette aux contraintes de l'estimé de l'ensemble actif  $\mathcal{A}$  (contraintes égalitaires), i.e., résoudre le système de KKT

$$\begin{pmatrix} H & A^T & \tilde{B}^T \\ A & 0 & 0 \\ \tilde{B} & 0 & 0 \end{pmatrix} \begin{pmatrix} -p \\ \tilde{\eta} \\ \tilde{\lambda} \end{pmatrix} = \begin{pmatrix} g \\ a \\ \tilde{b} \end{pmatrix},$$

où le vecteur  $\tilde{b}$  représente le sous-ensemble du vecteur  $b$  correspondant aux contraintes inégalitaires actives, et  $\tilde{B}$  représente la matrice jacobienne correspondante pour les contraintes de  $\mathcal{A}$ . Les multiplicateurs de Lagrange correspondant aux contraintes égalitaires sont notés  $\eta$ .

- 2 Choisir le pas<sup>15</sup> le plus grand possible dans la direction  $p$  ne violant aucune contrainte inégalitaire inactive, i.e.,

$$\bar{x} = x + \alpha p,$$

où la longueur du pas  $0 \leq \alpha \leq 1$  est choisie de manière à maintenir la faisabilité par rapport aux contraintes inégalitaires inactives.

- 3 **si** le pas est réduit, i.e.,  $\alpha < 1$  **alors**
- 4     Ajouter l'inégalité limitante à l'ensemble actif  $\mathcal{A}$ , et retourner à l'étape 1.
- 5 **sinon**
- 6     prendre le pas plein, i.e.,  $\alpha = 1$ , et vérifier le signe des multiplicateurs de Lagrange.
- 7     **si** toutes les inégalités possèdent des multiplicateurs  $\lambda$  positifs **alors**
- 8         **retourner**  $(x^*, \lambda^*, \eta^*)$ .
- 9     **sinon**
- 10         Enlever de l'ensemble actif  $\mathcal{A}$  la contrainte inégalitaire possédant le multiplicateur  $\lambda$  le plus négatif et retourner à l'étape 1.
- 11     **fin**
- 12 **fin**

Le choix de fonctions de mérite dans le cadre de l'optimisation sous contraintes est plus difficile, car il est nécessaire d'équilibrer les objectifs souvent conflictuels associés à la minimisation de la fonction de coût, et au respect des contraintes égalitaires/inégalitaires. On utilise alors des méthodes basées sur les *fonctions de pénalité*. Par exemple, une fonction de pénalité quadratique candidate est donnée par

$$P(x, \rho) = F(x) + \frac{\rho}{2} c^T(x) c(x),$$

<sup>15</sup>la notion de pas intervient dans les méthodes de recherche linéaire, nous en parlons plus bas.

où  $\rho$  représente le *paramètre de pénalité* ou la *pondération de pénalité*. Malheureusement, ce type de fonction de pénalité est mal conditionné, car l'accroissement du paramètre  $\rho$  engendre des problèmes d'optimisation sans contraintes toujours plus difficiles à résoudre. Une alternative consiste alors à utiliser une *fonction lagrangienne augmentée*, constituée du lagrangien (1.13) et d'un terme de pénalité sous la forme (1.27), telle que

$$P(x, \lambda, \rho) = L(x, \lambda) + \frac{\rho}{2} c^T(x) c(x) = F(x) - \lambda^T c(x) + \frac{\rho}{2} c^T(x) c(x).$$

2. Les *méthodes de recherche linéaire*, permettant de modifier la *longueur* du pas d'une itération par l'introduction d'un paramètre  $\alpha$  tel que

$$\bar{x} = x + \alpha p. \tag{1.28}$$

En utilisant l'expression (1.28), la nouvelle fonction de mérite associée peut s'écrire exclusivement en fonction du paramètre  $\alpha$ , telle que

$$M(\bar{x}) = M(x + \alpha p) = M(\alpha).$$

La plupart des algorithmes d'optimisation commencent avec le plein pas de Newton, i.e.,  $\alpha^{(0)} = 1$ . Les itérés suivants sont alors calculés jusqu'à ce qu'une longueur de pas  $\alpha^{(k)}$  satisfasse une condition de décroissance suffisante donnée par la formule de Goldstein-Armijo [41]

$$0 < -\kappa_1 \alpha^{(k)} M'(0) \leq M(0) - M(\alpha^{(k)}) \leq -\kappa_2 \alpha^{(k)} M'(0),$$

où  $M'(0) = (\nabla M(0))^T p$  représente la dérivée directionnelle pour  $\alpha = 0$ , et où les constantes  $\kappa_1$  et  $\kappa_2$  vérifient  $0 < \kappa_1 \leq \kappa_2 < 1$ .

3. Les *méthodes à régions de confiance*, permettant d'ajuster à la fois la direction de recherche  $p$  et la longueur du pas de Newton  $\alpha$  pour définir une itération du type  $\bar{x} = x + \alpha p$ . Supposons que  $x$  soit fixé, et que nous cherchons à déterminer les valeurs des variables  $p$  permettant de minimiser

$$F(x + p) \approx F(x) + g^T p + \frac{1}{2} p^T H p,$$

sous les contraintes

$$\frac{1}{2} p^T p \leq \delta^2.$$

Nous considérons que l'on peut se fier à la prédiction  $F(x + p)$  si les points obtenus appartiennent à la région définie par le *rayon de confiance*  $\delta$ . Le lagrangien associé à ce problème est défini par

$$L_T(p, \tau) = F(x) + g^T p + \frac{1}{2} p^T H p - \tau \left( \delta^2 - \frac{1}{2} p^T p \right),$$

où  $\tau$  représente le multiplicateur de Lagrange associé aux contraintes inégalitaires de la région de confiance. Par conséquent, la condition nécessaire d'optimalité s'écrit

$$\nabla_p L_T(p, \tau) = g + H p + \tau p = g + (H + \tau I) p = 0.$$

Il y a deux solutions possibles à ce problème : soit la contrainte associée au rayon de confiance est inactive, et alors  $\tau = 0$  et la direction de recherche  $p$  est la direction de Newton classique ; soit la contrainte du rayon de confiance est active, et alors  $\|p\| = \delta$  et  $\tau > 0$ . Le rayon de confiance  $\delta$  et le multiplicateur  $\tau$  sont liés : lorsque  $\delta$  est grand, alors  $\tau$  est petit, et vice versa.

4. Les *techniques de filtrage*, permettant d'éviter le réglage souvent délicat d'un paramètre de pénalité dans une fonction de mérite. L'approche par filtrage permet d'accepter une nouvelle itération de Newton s'il existe un point  $x$  minimisant soit la fonction de coût  $F(x)$ , soit la *violation* des contraintes  $v(c(x))$ , où la violation est mesurée par une norme appropriée. L'idée est alors de comparer les informations contenues dans l'itération courante, à celles contenues dans les itérations précédentes, afin de "filtrer" les mauvais itérés. Considérons que les valeurs de la fonction de coût et de la violation de contraintes au point  $x^{(k)}$  soient données par

$$\{F^{(k)}, v^{(k)}\} \equiv \{F(x^{(k)}), v(c(x^{(k)}))\}.$$

Lorsque l'on compare les informations en deux points différents  $x^{(k)}$  et  $x^{(j)}$ , on dit qu'une paire  $\{F^{(k)}, v^{(k)}\}$  *domine* une autre paire  $\{F^{(j)}, v^{(j)}\}$  si et seulement si  $F^{(k)} < F^{(j)}$  et  $v^{(k)} < v^{(j)}$ . Avec cette définition, il est alors possible de construire un *filtre* comme une liste de paires

$$\mathcal{F} = \begin{pmatrix} F^{(1)}, v^{(1)} \\ F^{(2)}, v^{(2)} \\ \vdots \\ F^{(K)}, v^{(K)} \end{pmatrix},$$

tel qu'aucune paire ne domine les autres. On dit alors qu'un nouveau point  $\{F^{(l)}, v^{(l)}\}$  est *acceptable* pour être inclu dans le filtre s'il n'est dominé par aucun autre point du filtre.

## Programmation non linéaire

Le problème de programmation non linéaire (NLP) [34] consiste à déterminer un vecteur  $x = (x_1, \dots, x_n)$ , permettant de minimiser la fonction objectif scalaire

$$F(x),$$

sujette aux  $m$  contraintes

$$c_L \leq c(x) \leq c_U, \tag{1.29}$$

et aux bornes simples

$$x_L \leq x \leq x_U. \tag{1.30}$$

Des contraintes égalitaires peuvent alors être imposées en posant  $c_L = c_U$ .

Les conditions KKT nécessaires pour que le point  $X^*$  soit une solution optimale du problème NLP requièrent que

- $X^*$  soit une solution admissible, i.e., (1.29) et (1.30) soient satisfaites,
- Les multiplicateurs de Lagrange  $\lambda$  associés à (1.29) et  $\nu$  associés à (1.30) satisfassent

$$g = G^T \lambda + \nu,$$

- les multiplicateurs de Lagrange associés aux contraintes inégalitaires soient

$$\begin{cases} \textit{non positifs} & \text{pour les frontières supérieures actives,} \\ \textit{nuls} & \text{pour les contraintes identiquement satisfaites,} \\ \textit{non négatifs} & \text{pour les frontières inférieures actives.} \end{cases}$$

- La matrice jacobienne  $\tilde{G}$  correspondant aux contraintes actives soit de plein rang ligne.

Les algorithmes de *programmation quadratique séquentielle* (SQP) font sans nul doute partie des méthodes les plus répandues pour résoudre les problèmes de type NLP. L'idée principale de la méthode d'optimisation sous contraintes égalitaires, décrite dans la section 1.2.3, était d'approximer le lagrangien sous forme quadratique et les contraintes de manière linéaire (cf équations (1.22) et (1.23)). Dans le cadre de l'optimisation non linéaire NLP par approche SQP décrite ici, l'idée est de résoudre de manière itérative un sous-problème QP. Supposons que l'itération débute au point  $(x, \lambda, \nu)$ . Le sous-problème QP revient alors déterminer la direction de recherche  $p$  permettant de minimiser l'approximation quadratique du lagrangien

$$g^T p + \frac{1}{2} p^T H p,$$

sujet aux approximations linéaires des contraintes

$$b_L \leq \begin{pmatrix} Gp \\ p \end{pmatrix} \leq b_U,$$

où les bornes inférieure et supérieure sont définies par

$$b_L = \begin{pmatrix} c_L - c \\ x_L - x \end{pmatrix}, \quad b_U = \begin{pmatrix} c_U - c \\ x_U - x \end{pmatrix}.$$

La solution de ce problème QP ne définit pas seulement la direction de recherche  $p$ , mais l'algorithme d'optimisation quadratique détermine également un estimé de l'ensemble actif des contraintes, ainsi qu'un estimé des multiplicateurs de Lagrange  $\hat{\lambda}$  et  $\hat{\nu}$  associés. Ainsi, il est possible de définir des directions de recherche pour les multiplicateurs sous la forme

$$\begin{aligned} \Delta\lambda &\equiv \hat{\lambda} - \lambda, \\ \Delta\nu &\equiv \hat{\nu} - \nu. \end{aligned}$$

Finalement, la solution du sous-problème QP peut être utilisée afin de construire l'information nécessaire au calcul d'une fonction de mérite. Une prédiction linéaire de la valeur des contraintes est donnée par

$$\bar{s} \equiv Gp + c.$$

En supposant que l'on débute l'itération en ayant un estimé des *variables de relaxation*  $s$ , alors il vient

$$\Delta s \equiv \bar{s} - s = Gp + (c - s). \quad (1.31)$$

Le terme  $(c - s)$  dans l'équation (1.31) permet de mesurer la "déviation de la linéarité" dans les contraintes  $c$ . Par exemple, si les contraintes  $c(x)$  sont des fonctions linéaires des variables  $x$ , alors  $(c - s) = 0$ . Maintenant, la direction de recherche augmentée, formée en regroupant les expressions précédentes, s'écrit

$$\begin{pmatrix} \bar{x} \\ \bar{\lambda} \\ \bar{\nu} \\ \bar{s} \end{pmatrix} = \begin{pmatrix} x \\ \lambda \\ \nu \\ s \end{pmatrix} + \alpha \begin{pmatrix} p \\ \Delta\lambda \\ \Delta\nu \\ \Delta s \end{pmatrix},$$

où  $\alpha$  représente la longueur du pas. Lorsqu'un pas de Newton plein est utilisé ( $\alpha = 1$ ), alors les nouveaux multiplicateurs de Lagrange du problème NLP deviennent ceux du problème QP, i.e.,  $\bar{\lambda} = \hat{\lambda}$  et  $\bar{\nu} = \hat{\nu}$ .

Comme pour les autres algorithmes d'optimisation basés sur la méthode de Newton, il est nécessaire de définir une stratégie de globalisation pour le problème NLP. Une forme modifiée de la fonction de mérite lagrangienne augmentée peut alors être choisie, telle que

$$M(x, \lambda, s) = F - \lambda^T(c - s) + \frac{1}{2}(c - s)^T \Theta (c - s),$$

où  $\Theta$  est une matrice diagonale de paramètres de pénalité vérifiant

$$\frac{\partial M}{\partial x}(0) \leq -\frac{1}{2}p^T H p.$$

### Quelques solveurs NLP classiques

Devant le nombre très important de solveurs de programmation non linéaire existant, nous ne cherchons pas ici à les décrire tous de manière exhaustive, mais seulement à présenter brièvement quelques outils logiciels classiques, largement utilisés dans la communauté de l'optimisation<sup>16</sup>.

**Fmincon** [42] : Ce solveur fait partie intégrante de la boîte à outils "Optimisation" de l'environnement de calcul Matlab. Il utilise un algorithme de type SQP. La formule BFGS est utilisée afin de maintenir une approximation de la matrice hessienne. Une fonction de mérite de type Han est utilisée afin de déterminer la longueur du pas d'optimisation. Cet algorithme d'optimisation nécessite beaucoup d'itérations avant de converger, et a été dépassé depuis quelques années par d'autres solveurs comme NPSOL ou SNOPT, utilisant des méthodes de calcul plus abouties.

**SNOPT** [43] : Développé et maintenu par des chercheurs de l'Université de Californie,

<sup>16</sup>il existe néanmoins de nombreux autres solveurs tels que CONOPT, LANCELOT, OPTIMA, SOCS, LOQO, ...



San Diego et de l'Université de Stanford (Stanford Systems Optimization Laboratory : SOL), SNOPT est un solveur dédié aux problèmes d'optimisation sous contraintes (LP, QP et NLP). Dans le cas des problèmes NLP, SNOPT utilise une méthode de type SQP "creuse", basée sur des approximations de type quasi-Newton du hessien de la fonction lagrangienne (rafraîchissement du type BFGS). La fonction de mérite utilisée est une fonction augmentée, comme c'est aussi le cas pour le solveur SQP "dense" NPSOL (voir ci-dessous). De manière générale, SNOPT requiert moins de calculs matriciels que NPSOL pour aboutir à une solution, et moins d'évaluations de fonctions que l'algorithme non-linéaire MINOS (voir plus bas). SNOPT exploite les propriétés des matrices creuses et est, par conséquent, bien adapté pour traiter les problèmes de grande dimension comprenant plusieurs centaines de variables et de contraintes.

**NPSOL** [44] : Comme SNOPT, NPSOL est un solveur développé et maintenu par des chercheurs de l'Université de Californie, San Diego et de l'Université de Stanford. Les algorithmes développés pour les deux solveurs sont sensiblement les mêmes. NPSOL est un algorithme SQP de type "dense" (matrices pleines) et, par conséquent, bien adapté aux problèmes de petite et moyenne taille, bien qu'il n'existe pas, a proprement parler, de limite sur le nombre de variables/contraintes pouvant intervenir dans le problème. NPSOL inclut également le solveur de problèmes d'optimisation quadratiques LLSOL.

**MINOS** [45] : Solveur également développé par des chercheurs de l'Université de Californie, San Diego et de l'Université de Stanford, MINOS est un solveur dédié aux problèmes de type LP, QP et NLP de grandes dimensions. L'algorithme d'optimisation SQP "creux" développé pour MINOS est quelque peu différent de celui utilisé par les solveurs SNOPT et NPSOL, la principale différence se situant au niveau de la fonction lagrangienne, quadratique pour SNOPT et NPSOL, non linéaire pour MINOS. Ce solveur est surtout recommandé si les contraintes ou la fonction objectif sont relativement faciles à évaluer. Si les contraintes sont faiblement non linéaires, il y a de bonnes chances pour qu'une solution soit obtenue, mais sans garantie pour autant. Si certaines contraintes sont fortement non linéaires et/ou leurs gradients difficiles à évaluer, il vaut mieux alors se tourner vers des solveurs comme SNOPT ou NPSOL.

**filterSQP** [46] : C'est un algorithme de type SQP, développé par l'Université de Dundee (UK) et basé sur les méthodes d'optimisation par régions de confiance. Il comprend de plus un mécanisme de "filtrage" afin de promouvoir la convergence globale du problème. Le rayon de la région de confiance est adapté par le solveur durant la résolution du problème. filterSQP fonctionne autant pour les problèmes de type "dense" que "creux".

**KNITRO** [47] : Cette librairie logicielle récente fournit trois solveurs permettant de résoudre des problèmes NLP : un algorithme direct du point intérieur qui applique les techniques de barrières et factorise directement la matrice KKT du système non singulier, un algorithme du point intérieur qui applique les techniques de barrières en utilisant la méthode du gradient conjugué afin de résoudre les sous-problèmes KKT, et un algorithme basé sur le concept d'ensembles actifs. KNITRO est dédié aux problèmes de type LP, QP, NLP de grandes dimensions, convexes ou non convexes. KNITRO est apprécié pour sa

robustesse et son efficacité.

### 1.3 Commande optimale

Fondamentalement, un problème d'optimisation de trajectoires peut être formulé comme un problème de commande optimale. Cette section se focalise sur la résolution de tels problèmes. Considérons un système commandé explicite, régi par des équations différentielles ordinaires, tel que

$$\dot{x}(t) = f(x(t), u(t), p, t), \quad (1.32)$$

où  $t$  représente la variable temporelle,  $x$  l'état du système,  $u$  les entrées de commande, et  $p$  un vecteur de paramètres.

**Définition 1.1** (Problème de commande optimale). *Le problème de commande optimale (OCP), consiste à déterminer les profils des entrées de commande  $u(t)$  (et/ou des paramètres  $p$ ) minimisant la fonctionnelle de coût*

$$\phi_0(x(t_0), u(t_0)) + \int_{t_0}^{t_f} \phi_t(x(t), u(t)) dt + \phi_f(x(t_f), u(t_f)), \quad (1.33)$$

et permettant

1. de vérifier la dynamique du système (1.32),
2. de satisfaire à un ensemble de contraintes :
  - initiales, à l'instant  $t_0$ , telles que

$$\psi_{0l} \leq \psi(x(t_0), u(t_0), p, t_0) \leq \psi_{0u}, \quad (1.34)$$

où  $\psi_0(x(t_0), u(t_0), p, t_0) \equiv \psi_0$ .

- finales, à l'instant  $t_f$ , définies par

$$\varphi_{fl} \leq \varphi(x(t_f), u(t_f), p, t_f) \leq \varphi_{fu}, \quad (1.35)$$

où  $\varphi_f(x(t_f), u(t_f), p, t_f) \equiv \varphi_f$ .

- de trajectoire, pour tout  $t_0 \leq t \leq t_f$ , telles que

$$g_l \leq g(x(t), u(t), p, t) \leq g_u. \quad (1.36)$$

- sur l'état, telles que

$$x_l \leq x(t) \leq x_u.$$

- sur les commandes, telles que

$$u_l \leq u(t) \leq u_u.$$

**Remarque 1.2.** La fonction objectif (1.33) est écrite ici sous la forme dite de Bolza. Si le terme intégral est nul, c'est à dire si la fonctionnelle de coût ne comprend que des quantités évaluées aux instants  $t_0$  et  $t_f$ , alors on dit qu'elle est sous la forme de Mayer. Si la fonctionnelle comprend uniquement un terme intégral, alors on dit qu'elle est sous la forme de Lagrange. Notons que la forme de Mayer peut être obtenue soit à partir de la forme de Bolza ou de Lagrange, en introduisant une variable d'état supplémentaire.

Contrairement aux problèmes de type NLP, qui comportent un ensemble fini de variables  $x$  et de contraintes  $c$ , les problèmes de commande optimale contiennent souvent des variables continues, telles que  $x(t)$  et  $u(t)$ , que nous cherchons à optimiser. De ce fait, un problème de commande optimale peut être vu comme une extension en dimension infinie d'un problème de programmation non linéaire.

### 1.3.1 Contraintes dynamiques

Supposons que l'on souhaite optimiser les entrées de commande  $u(t)$  minimisant le critère

$$J = \phi(x(t_f), t_f),$$

sujet à la dynamique

$$\dot{x}(t) = f(x(t), u(t)) \quad (1.37)$$

et aux contraintes de bord

$$\psi(x(t_f), u(t_f), t_f) = 0, \quad (1.38)$$

où les conditions initiales  $x(t_0) = x_0$  sont définies à l'instant initial  $t_0$  fixé, et où l'instant final  $t_f$  est libre. Ce problème introductif simple, appelé *forme autonome*, comporte deux types de contraintes égalitaires – des contraintes “continues” (1.37), qui doivent être satisfaites pour tout  $t_0 \leq t \leq t_f$ , et des contraintes “discrètes”, devant être vérifiées à l'instant  $t_f$  uniquement. De la même façon que nous avons défini un lagrangien (1.13), nous introduisons ici un indice de performance augmenté sous la forme

$$\hat{J} = [\phi + \nu^T \psi]_{t_f} - \int_{t_0}^{t_f} \lambda^T(t) \{\dot{x} - f(x(t), u(t))\} dt. \quad (1.39)$$

L'expression (1.39) contient des multiplicateurs de Lagrange  $\nu$  associés aux contraintes discrètes, mais également des multiplicateurs  $\lambda(t)$ , appelés *états adjoints* ou *co-états*, associés aux contraintes continues engendrées par les équations dynamiques (1.37) du système. En dimension finie, les conditions nécessaires d'optimalité (1.14) et (1.15) avaient été obtenues dans la section 1.2 en annulant les dérivées premières du lagrangien. Dans le cas continu, l'opération analogue consiste à annuler la *variation première*  $\delta \hat{J} = 0$ . On définit pour cela l'hamiltonien

$$H = \lambda^T(t) f(x(t), u(t)), \quad (1.40)$$

et la fonction auxiliaire

$$\Phi = \phi + \nu^T \psi.$$

Les conditions nécessaires d'optimalité, appelées *équations d'Euler-Lagrange*, sont alors données par :

les *équations adjointes*

$$\dot{\lambda} = -H_x^T, \quad (1.41)$$

les *équations de commande*

$$0 = H_u^T, \quad (1.42)$$

et les conditions de transversalité

$$\lambda(t_f) = \Phi_x^T|_{t=t_f}, \quad (1.43)$$

$$0 = (\Phi_t + H)|_{t=t_f}, \quad (1.44)$$

$$0 = \lambda(t_0). \quad (1.45)$$

Dans ces expressions, les dérivées partielles  $H_x$ ,  $H_u$  et  $\Phi_x$  sont des vecteurs lignes, i.e.,  $H_x \equiv (\frac{\partial H}{\partial x_1}, \dots, \frac{\partial H}{\partial x_n})$ . Les équations de commande (1.41) représentent une version simplifiée du *principe du maximum de Pontryagin*<sup>17</sup>, dont la forme générale s'écrit

$$u = \underset{u \in \mathcal{U}}{\operatorname{argmin}} H, \quad (1.46)$$

où  $\mathcal{U}$  représente le domaine admissible des commandes. Le principe du maximum affirme que les variables de commande doivent être choisies de manière à optimiser, à tout instant, l'hamiltonien, sujet à des limitations sur la commande imposées par les contraintes de trajectoire sur l'état et sur la commande. L'ensemble complet des conditions nécessaires d'optimalité constitue un système d'équations algébriques différentielles (DAE) formé par les équations (1.37), (1.41), (1.42), et avec les conditions de bord (1.43), (1.44), (1.45) et (1.38) aux instants  $t_0$  et  $t_f$ . Cette formulation est communément appelée *problème aux deux bouts*.

La résolution des conditions d'optimalité issues du principe du maximum est donc un ingrédient nécessaire à la résolution d'un problème aux deux bouts non contraint. Dans la section suivante, nous montrons comment tenir compte de contraintes égalitaires et inégalitaires dans le problème aux deux bouts.

### 1.3.2 Contraintes égalitaires algébriques

Nous généralisons le problème d'optimisation de la section précédente, en introduisant des contraintes de trajectoire algébriques du type

$$0 = g(x(t), u(t), t), \quad (1.47)$$

qui viennent s'ajouter aux contraintes dynamiques (1.37) et aux contraintes de bord (1.38). Le traitement des contraintes de trajectoire 1.47 dépend de la matrice jacobienne

$$g_u = \begin{pmatrix} \frac{\partial g_1}{\partial u_1} & \frac{\partial g_1}{\partial u_2} & \dots & \frac{\partial g_1}{\partial u_n} \\ \frac{\partial g_2}{\partial u_1} & \frac{\partial g_2}{\partial u_2} & \dots & \frac{\partial g_2}{\partial u_n} \\ \vdots & & \ddots & \\ \frac{\partial g_n}{\partial u_1} & \frac{\partial g_n}{\partial u_2} & \dots & \frac{\partial g_n}{\partial u_m} \end{pmatrix}.$$

<sup>17</sup>bien que Pontryagin l'ait historiquement défini comme un problème de maximisation [48], il est aussi d'usage courant de l'appeler le principe du minimum de Pontryagin, car l'hamiltonien utilisé dans la littérature est souvent négligé par rapport à l'hamiltonien de Pontryagin.

Dans ce cas, deux situations peuvent se présenter. Si la matrice  $g_u$  est de plein rang, alors le système d'équations différentielles et algébriques défini par (1.37) et (1.47) est un système DAE d'ordre 1, et la fonction hamiltonienne (1.40) peut être remplacée par

$$H = \lambda^T f + \mu^T g, \quad (1.48)$$

qui va modifier à son tour les équations adjointes (1.41) et les équations de commande (1.42).

Si la matrice  $g_u$  présente une chute de rang, alors on peut différencier l'équation (1.47) par rapport au temps, ce qui donne

$$\begin{aligned} 0 &= g_x \dot{x} + g_u \dot{u} + g_t \\ &= g_x f(x(t), u(t)) + g_u \dot{u} + g_t \\ &= g'(x(t), u(t), t). \end{aligned}$$

Nous obtenons ainsi une nouvelle contrainte de trajectoire  $g'$  algébriquement équivalente à la contrainte initiale. Pour cette nouvelle contrainte de trajectoire, il faut vérifier à nouveau le rang de la matrice  $g'_u$ . Si la matrice est de plein rang, alors on dit que le système DAE initial est d'ordre 2. Dans le cas contraire, s'il y a une chute de rang, il faut alors redéfinir la fonction hamiltonienne (1.48) en remplaçant  $g$  par  $g'$ , et réitérer le processus. Cette approche est connue sous le nom de *réduction d'ordre* dans le cadre des équations algébro-différentielles [49].

**Remarque 1.3.** Nous avons considéré ici un système DAE apparaissant lorsque des contraintes de trajectoires sont imposées dans le problème de commande optimale, telles que

$$\begin{aligned} \dot{x}(t) &= f(x(t), u(t), t), \\ 0 &= g(x(t), u(t), t). \end{aligned}$$

Cependant, dans le cas d'une absence de contraintes de trajectoire, les conditions nécessaires d'optimalité (1.37), (1.41) et (1.42) conduisent à un système DAE sous la forme

$$\dot{x}(t) = f(x(t), u(t)), \quad (1.49)$$

$$\dot{\lambda}(t) = -H_x^T, \quad (1.50)$$

$$0 = H_u^T, \quad (1.51)$$

où les variables différentielles sont  $(x, \lambda)$  et les variables algébriques sont données par  $u$ . Lorsque la matrice  $H_{uu}$  n'est pas singulière<sup>18</sup>, la commande  $u$  est obtenue via la condition d'optimalité  $0 = H_u^T$ . Dans le cas contraire, la commande n'est pas définie de manière unique par la condition d'optimalité 1.51. Cette situation est communément appelée un *arc singulier*, et son analyse repose sur des techniques assez similaires à celles introduites précédemment pour la prise en compte de contraintes de trajectoire.

---

<sup>18</sup>la matrice  $H_{uu}$  est définie par  $H_{uu} = \frac{\partial H}{\partial u_i \partial u_j} \Big|_{i,j=1,\dots,m}$ , où  $m$  représente le nombre de commandes du système, i.e.,  $m = \dim u$ .

### 1.3.3 Contraintes inégalitaires algébriques

Considérons maintenant des contraintes inégalitaires de trajectoire sous la forme

$$0 \leq g(x(t), u(t), t).$$

Contrairement aux contraintes égalitaires, qui doivent être vérifiées pour tout  $t_0 \leq t \leq t_f$ , les contraintes inégalitaires peuvent être soit actives ( $0 = g$ ) ou inactives ( $0 < g$ ) à chaque instant, i.e., le domaine temporel est partitionné en arcs contraints et non contraints. Durant les arcs non contraints, les conditions nécessaires d'optimalité sont données par (1.37), (1.41) et (1.42), alors que les conditions d'optimalité avec l'hamiltonien modifié (1.48) sont applicables pour les arcs contraints.

Ainsi, la prise en compte de contraintes inégalitaires engendre trois complications majeures :

1. le nombre d'arcs contraints présent dans la solution optimale n'est pas connu *a priori*,
2. la position des points de jonction entre les arcs contraints et les arcs non contraints n'est pas connue,
3. aux points de jonction, il est possible que les variables de commande  $u$  et les états adjoints  $\lambda$  soient tous deux discontinus. Dans ce cas, des *conditions de saut*<sup>19</sup> additionnelles doivent être satisfaites [34].

## 1.4 Propagation de trajectoires

Les *techniques de propagation de trajectoires*, également appelées *problèmes aux valeurs initiales* (IVP) ou *méthodes d'intégration numérique*, sont essentielles pour un bon nombre d'algorithmes d'optimisation de trajectoires, comme par exemple les méthodes de tirs simples ou multiples. Les algorithmes numériques permettant de résoudre les problèmes de type IVP sont relativement matures en comparaison avec d'autres champs applicatifs de la commande optimale. La solution du problème aux valeurs initiales permet d'intégrer numériquement un ensemble d'équations différentielles ordinaires (ODE). Ainsi, le problème IVP consiste à déterminer la valeur de  $x(t_f)$  à un instant  $t_0 < t_f$ , vérifiant la dynamique

$$\dot{x}(t) = f(x(t), t), \tag{1.52}$$

en partant de la condition initiale  $x(t_0) = x_0$ .

la plupart des méthodes de propagation de trajectoires peuvent être regroupées en deux grandes catégories :

- les méthodes *à pas unique*,
- les méthodes *multipas*.

---

<sup>19</sup>qui sont des conditions de bord imposées aux points de jonction.

Commençons par traiter le cas de la propagation d'ODE en un seul pas, par exemple entre  $t_i$  et  $t_{i+1}$ . En notant  $x(t_i) \equiv x_i$ , l'intégration formelle de la dynamique (1.52) donne

$$x_{i+1} = x_i + \int_{t_i}^{t_{i+1}} \dot{x}(t) dt = x_i + \int_{t_i}^{t_{i+1}} f(x(t), t) dt. \quad (1.53)$$

Afin d'approximer l'intégrale par une somme discrète, divisons le pas d'intégration en  $k$  sous-intervalles

$$\tau_j = t_i + h_i \rho_j, \quad (1.54)$$

avec  $0 \leq \rho_1 \leq \rho_2 \leq \dots \leq \rho_k \leq 1$ , pour tout  $1 \leq j \leq k$ , où  $h_i$  représente le *pas d'intégration*. Sur ce domaine partitionné, il est alors possible d'approximer l'intégrale de (1.53) par la loi de quadrature

$$\int_{t_i}^{t_{i+1}} f(x(t), t) dt \approx h_i \sum_{j=1}^k \beta_j \hat{f}_j,$$

où  $\hat{f}_j \equiv f(\tau_j, \hat{x}_j)$ . Les valeurs des variables  $x$  aux points intermédiaires  $\tau_j$ , telles que  $\hat{x}(\tau_j) \equiv \hat{x}_j$ , sont alors construites en utilisant l'expression

$$\int_{t_i}^{\tau_j} f(x(t), t) dt \approx h_i \sum_{l=1}^k \alpha_{jl} f_l,$$

pour  $1 \leq j \leq k$ .

En regroupant ces résultats, on retrouve un ensemble de techniques classiques de propagation à pas unique, appelées *méthodes de Runge-Kutta à  $k$  étages*<sup>20</sup>, telles que

$$x_{i+1} = x_i + h_i \sum_{j=1}^k \beta_j f_{ij},$$

où

$$f_{ij} = f \left[ \left( x_i + h_i \sum_{l=1}^k \alpha_{jl} f_{il} \right), (t_i + h_i \rho_j) \right], \quad 1 \leq j \leq k.$$

Dans ces expressions,  $\{\rho_j, \beta_j, \alpha_{jl}\}$  représente des constantes connues, avec  $0 \leq \rho_1 \leq \rho_2 \leq \dots \leq 1$ . Une manière commode de définir ces coefficients est de les représenter sous la forme de tableaux de Butcher

$$\begin{array}{c|ccc} \rho_1 & \alpha_{11} & \dots & \alpha_{1k} \\ \vdots & \vdots & & \vdots \\ \rho_k & \alpha_{k1} & \dots & \alpha_{kk} \\ \hline & \beta_1 & \dots & \beta_k \end{array}$$

Les schémas d'intégration numérique d'ODE sont dits *explicites* si  $\alpha_{jl} = 0$  pour  $l \geq j$  et *implicites* autrement. Les quatre méthodes de Runge-Kutta à  $k$  étages les plus répandues

---

<sup>20</sup> $k$  est couramment appelé l'"étage".

sont illustrées ci-dessous. Dans chaque cas, nous donnons le tableau de Butcher correspondant, ainsi que la représentation discrète associée.

**Méthode d'Euler** (explicite,  $k = 1$ )

Tableau de Butcher :

$$\begin{array}{c|c} 0 & 0 \\ \hline & 1 \end{array}$$

Représentation discrète :

$$x_{i+1} = x_i + h_i f_i. \tag{1.55}$$

**Méthode classique de Runge-Kutta (RK4)** (explicite,  $k = 4$ )

Tableau de Butcher :

$$\begin{array}{c|cccc} 0 & 0 & 0 & 0 & 0 \\ 1/2 & 1/2 & 0 & 0 & 0 \\ 1/2 & 0 & 1/2 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 \\ \hline & 1/6 & 1/3 & 1/3 & 1/6 \end{array}$$

Représentation discrète :

$$\begin{aligned} k_1 &= h_i f_i, \\ k_2 &= h_i f\left(x_i + \frac{1}{2}k_1, t_i + \frac{h_i}{2}\right), \\ k_3 &= h_i f\left(x_i + \frac{1}{2}k_2, t_i + \frac{h_i}{2}\right), \\ k_4 &= h_i f(x_i + k_3, t_{i+1}), \\ x_{i+1} &= x_i + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4). \end{aligned} \tag{1.56}$$

**Méthode trapézoïdale** (implicite,  $k = 2$ )

Tableau de Butcher :

$$\begin{array}{c|cc} 0 & 0 & 0 \\ 1 & 1/2 & 1/2 \\ \hline & 1/2 & 1/2 \end{array}$$

Représentation discrète :

$$x_{i+1} = x_i + \frac{h_i}{2}(f_i + f_{i+1}). \tag{1.57}$$

**Méthode de Hermite-Simpson** (implicite,  $k = 3$ )

Tableau de Butcher :

$$\begin{array}{c|ccc} 0 & 0 & 0 & 0 \\ 1/2 & 5/24 & 1/3 & -1/24 \\ 1 & 1/6 & 2/3 & 1/6 \\ \hline & 1/6 & 2/3 & 1/6 \end{array}$$



Représentation discrète :

$$\begin{aligned}\bar{x} &= \frac{1}{2}(x_i + x_{i+1}) + \frac{h_i}{8}(f_i - f_{i+1}), \\ \bar{f} &= f_i \left( \bar{x}, t_i + \frac{h_i}{2} \right), \\ x_{i+1} &= x_i + \frac{h_i}{6}(f_i + 4\bar{f} + f_{i+1}).\end{aligned}\tag{1.58}$$

La méthode de Runge-Kutta est une méthode dite de *collocation*. En effet, supposons que l'on désire approximer la dynamique (1.52) sous la forme d'une fonction polynomiale  $\tilde{x}(t)$  de degré  $k$  sur chaque intervalle  $t_i \leq t \leq t_{i+1}$ , telle que

$$\tilde{x}(t) = a_0 + a_1(t - t_i) + \dots + a_k(t - t_i)^k,\tag{1.59}$$

où les coefficients  $a_0, a_1, \dots, a_k$  sont choisis de manière à ce que l'approximation "colle" aux données au début du pas  $t_i$ , i.e.,

$$\tilde{x}(t_i) = x_i,$$

et dont les dérivées premières vérifient les *conditions de collocation* aux points (1.54), i.e.,

$$\frac{d\tilde{x}(\tau_j)}{dt} = f(x(\tau_j), \tau_j).$$

**Remarque 1.4.** (a) Nous avons choisi ici de représenter la solution de la méthode d'intégration numérique de Runge-Kutta sous la forme d'un polynôme par morceaux (1.59). Dans le cas général, nous utiliserons des polynômes continus par morceaux appelés B-splines, possédant de meilleures propriétés numériques (voir annexe B).

(b) De nombreuses autres stratégies de collocation existent. Par exemple, la méthode trapézoïdale et de Hermite-Simpson sont appelées *méthodes de Lobatto* : la première est basée sur un polynôme d'interpolation quadratique, et la seconde sur un interpolant cubique.

(c) La principale caractéristique des techniques explicites est que le calcul de chaque pas d'intégration peut être réalisé sans itération, voir par exemple (1.55) et (1.56). A l'inverse, les méthodes implicites nécessitent de résoudre une équation non linéaire du type

$$0 = x_{i+1} - x_i - \frac{h_i}{2}(f(x_{i+1}, t_{i+1}) + f(x_i, t_i)) \equiv \zeta_i.\tag{1.60}$$

Plus précisément, l'objectif est d'annuler le *résidu*  $\zeta_i$  en utilisant des étapes de *correction* et de *prédiction*.

La deuxième catégorie de méthodes d'intégration, appelées *méthodes multipas*, ont la forme générale suivante :

$$x_{i+k} = \sum_{j=0}^{k-1} \alpha_j x_{i+j} + h \sum_{j=0}^k \beta_j f_{i+j},\tag{1.61}$$

où  $\alpha_j$  et  $\beta_j$  sont des constantes connues. Si  $\beta_j = 0$ , alors la méthode est explicite ; autrement, elle est implicite. Ainsi, les *stratégies de Adams*, basées sur une approximation des fonctions  $f$  par des polynômes d'interpolation, font partie des méthodes d'intégration multipas. Ces méthodes engendrent quelques nouvelles difficultés, absentes avec les méthodes d'intégration à pas unique :

1. En observant l'équation (1.61), les méthodes d'intégration multipas utilisent les informations disponibles aux  $k - 1$  points précédents afin de générer un nouvel itéré. Ceci implique l'utilisation d'une technique appropriée afin d'initialiser le processus itératif. Par exemple, les premières itérations peuvent être effectuées par une méthode à pas unique.
2. Si le pas d'intégration  $h$  ou l'ordre  $k$  sont autorisés à varier dans l'expression (1.61), alors il faut veiller à ce que le calcul des coefficients soit efficace, et bien conditionné numériquement.

Le lecteur est renvoyé à l'ouvrage [34] pour obtenir plus de détails sur les algorithmes d'intégration multipas.

## 1.5 Méthodes classiques d'optimisation de trajectoires

La plupart des méthodes d'optimisation de trajectoires reposent sur une application de la méthode d'optimisation de Newton. Chacune d'entre-elles possède son propre mécanisme de traitement de la fonction de coût et des contraintes du problème de commande optimale. Les méthodes d'optimisation de trajectoire les plus courantes comprennent les techniques de tirs, directes ou indirectes, ainsi que les techniques de transcription, directes ou indirectes.

Les *techniques de tir* consistent à propager la dynamique du système en partant de conditions initiales données. L'optimisation consiste alors à déterminer les valeurs initiales d'un problème aux deux bouts.

Les *techniques de transcription* consistent à convertir le problème d'optimisation de dimension infinie en un problème de dimension finie. Les contraintes sont alors satisfaites en un nombre fini de points, appelés *points de collocation*.

Les *méthodes directes* transforment le problème de commande optimale en un problème d'optimisation paramétrique, dans lequel l'état et les commandes du système sont décrits à l'aide de courbes paramétrées. Les variables du problème d'optimisation sont alors les paramètres de dimensionnement de ces courbes, qui doivent vérifier les contraintes initiales, finales, ainsi que les contraintes de trajectoire. Les conditions d'optimalité sont alors imposées par l'algorithme de résolution NLP utilisé.

Les *méthodes indirectes* s'efforcent de résoudre de manière explicite les conditions d'optimalité issues du principe du maximum de Pontryagin. La résolution du problème de

commande optimale consiste alors à résoudre un problème aux deux bouts, pour lequel les variables d'optimisation deviennent les états adjoints, évalués à l'instant initial. Le comportement des commandes étant contraint par les conditions d'optimalité, l'objectif est alors de déterminer les conditions initiales de ces variables de décision permettant de rallier des conditions finales données.

### 1.5.1 Méthodes directes de tirs

#### Description de la méthode

Les techniques de tirs font partie des méthodes d'optimisation de trajectoire les plus courantes, et par là même les plus simples pour résoudre un problème aux deux bouts. Conceptuellement, la méthode de tir correspond à armer de façon correcte un "canon", de manière à ce qu'un boulet viennent frapper une cible donnée, d'où son nom. Dans notre cas, la trajectoire du boulet de canon revient à propager la dynamique du système de manière explicite (intégration numérique) entre deux instants  $t_0$  et  $t_f$ , l'armement du canon consiste à ajuster les conditions initiales du problème, et la cible correspond à un jeu de contraintes de bord à satisfaire à l'instant  $t_f$ . Un problème de programmation non linéaire est alors utilisé afin de déterminer la valeur initiale de la trajectoire. On considère que la trajectoire peut être décomposée en un ensemble fini de  $N$  segments de longueur variable, appelés *phases*<sup>21</sup>. Pour chaque phase, on définit alors les variables

$$X^{(k)} = \{x(t_0), p, t_0, x(t_f), t_f\}, \quad k = 1, \dots, N.$$

L'ensemble des variables du problème NLP est alors défini par

$$x \subset \{X^{(1)}, X^{(2)}, \dots, X^{(N)}\}.$$

De plus, les variables de commande doivent être définies par un ensemble fini de paramètres, par exemple sous la forme d'une fonction explicite telle que

$$u = p_1 + p_2 t,$$

ou d'une représentation implicite de la forme

$$0 = p_1 u(t) + \sin(p_2 u(t)).$$

Lorsque les commandes sont définies par une relation explicite, la propagation de la trajectoire peut être réalisée par la résolution d'un problème aux valeurs initiales IVP. En revanche, si les variables de commandes sont décrites de manière implicite, il faut alors utiliser une méthode d'intégration de type DAE, telle que décrite dans la section 1.3.1.

---

<sup>21</sup>Par exemple, dans le cas d'une mission de rentrée atmosphérique, on pourra considérer que la trajectoire globale peut être partitionnée en trois phases : la phase hypersonique, la phase de dissipation d'énergie en zone terminale (TAEM), et la phase d'atterrissage automatique (A&L).

La fonction de coût et les contraintes du problème de commande optimale, définies respectivement par (1.34) et (1.35), sont évaluées aux bords d'une ou plusieurs phases, i.e.,

$$c(x) = \begin{pmatrix} \psi^{(2)}(x^{(2)}(t_0^{(2)}), p^{(2)}, t_0^{(2)}) - \psi^{(1)}(x^{(1)}(t_f^{(1)}), p^{(1)}, t_f^{(1)}) \\ \vdots \\ \psi^{(N)}(x^{(N)}(t_0^{(N)}), p^{(N)}, t_0^{(N)}) - \psi^{(N-1)}(x^{(N-1)}(t_f^{(N-1)}), p^{(N-1)}, t_f^{(N-1)}) \end{pmatrix}. \quad (1.62)$$

En conclusion, la méthode directe de tirs peut être décrite par l'algorithme 4. La méthode

**Algorithme 4 : Méthode directe de tirs.**

**Entrées :** Variables de décision  $x$ .

**Sorties :** Fonction de coût  $F(x)$ , contraintes  $c(x)$  définies par (1.62).

**Algorithme :**

- 1 **tant que**  $c(x) \neq 0$  **faire**
- 2     **pour**  $k = 1, \dots, N$  **faire**
- 3         Initialiser la phase  $k$  :  $\{x^{(k)}(t_0^{(k)}), p^{(k)}, t_0^{(k)}\}$ .
- 4         Résoudre le problème aux valeurs initiales : calculer  $x^{(k)}(t_f^{(k)})$  en propageant la dynamique  $\dot{x} = f(x(t), t)$  de  $t_0^{(k)}$  à  $t_f^{(k)}$ .
- 5         Évaluer l'erreur sur la condition de bord
 
$$c(x^{(k)}) = \psi^{(k)}(x^{(k)}(t_0^{(k)}), p^{(k)}, t_0^{(k)}) - \psi^{(k-1)}(x^{(k-1)}(t_f^{(k-1)}), p^{(k-1)}, t_f^{(k-1)}).$$
- 6     **fin**
- 7     Ajuster les conditions initiales  $\{x^{(k)}(t_0^{(k)}), p^{(k)}, t_0^{(k)}\}$ ,  $k = 1, \dots, N$ , à l'aide d'une méthode d'optimisation de type NLP, de manière que  $c(x^{(k)}) = 0$ ,  $k = 1, \dots, N$ .
- 8 **fin**
- 9 Calculer la fonction de coût  $F(x)$ .
- 10 **retourner**  $(F(x), c(x))$ .

directe de tirs est largement utilisée de par le faible nombre de variables nécessaires pour décrire le problème d'optimisation. Cependant, si le profil des commandes ne peut être décrit par un petit nombre de variables, les performances associées à la méthode de tirs directs se dégradent de manière significative. Par exemple, si l'on choisit de décrire les variables de commande sous la forme

$$u(t) = \sum_{k=1}^M p_k B_k(t), \quad M \gg 1, \quad (1.63)$$

où  $B_k(t)$  représente un ensemble de fonctions de support, par exemple des B-splines, alors le calcul de l'information de gradient, nécessaire à l'application d'une méthode de résolution de type NLP, peut devenir très sensible aux conditions initiales. En effet, une petite variation de ces conditions peut entraîner des variations très importantes des conditions finales, du fait de la propagation des équations différentielles ordinaires, et du comportement hautement non linéaire des contraintes  $c(x)$  par rapport aux variables d'optimisation.

Nous verrons dans la section 1.5.3 que les méthodes de tirs multiples permettent notamment de corriger ce phénomène. De plus, le nombre important de paramètres utilisés dans l'expression (1.63) pour décrire les profils des commandes entraîne une charge importante de calculs nécessaires à la détermination de l'information de gradient.

### Applications à l'optimisation de trajectoires de rentrée atmosphérique

Les méthodes directes de tirs ont été largement utilisées pour l'optimisation de trajectoires de rentrée atmosphérique [50, 51, 52, 53] (voir le tableau 1.3 pour plus de références). Nous décrivons ici une méthode, proposée dans [54], pour optimiser des trajectoires d'atterrissage automatique pour le véhicule expérimental X-34.

Sous l'hypothèse "Terre plate", les équations dynamiques longitudinales régissant le mouvement du centre de gravité du véhicule par rapport à un repère cartésien local lié à la piste, sont données par [32]

$$\dot{x} = V \cos \gamma, \quad (1.64)$$

$$\dot{h} = V \sin \gamma, \quad (1.65)$$

$$\dot{V} = -\frac{D(\alpha, M)}{m} - g \sin \gamma, \quad (1.66)$$

$$\dot{\gamma} = \frac{L(\alpha, M)}{mV} - \frac{g}{V} \cos \gamma, \quad (1.67)$$

où  $x$  représente la portée longitudinale,  $h$  l'altitude,  $V$  la vitesse relative du véhicule,  $\gamma$  sa pente. Les termes  $L$  et  $D$  sont respectivement la force de portance et la force de traînée, telles que

$$L(\alpha, M) = \bar{q} S C_L(\alpha, M), \quad (1.68)$$

$$D(\alpha, M) = \bar{q} S C_D(\alpha, M), \quad (1.69)$$

où  $\bar{q} = \frac{1}{2}\rho V^2$  est la pression dynamique,  $S$  la surface aérodynamique de référence, et  $C_L$  et  $C_D$  les coefficients de portance et de traînée tabulés en configuration lisse.

Le profil de la trajectoire d'atterrissage, représenté sur la figure 1.1, est similaire à celui obtenu pour la navette spatiale américaine. La phase d'atterrissage débute au point d'entrée ALI (voir figure 1.1), lorsque le nombre de Mach est égal à 0.5 (à environ 5km d'altitude), et se termine lorsque le véhicule s'arrête sur la piste. La trajectoire comprend principalement trois segments :

1. une phase de descente parcourue avec une pente forte constante  $\gamma_1$ , permettant d'annuler les dispersions sur les conditions cinématiques initiales,
2. un arrondi circulaire de rayon  $R$  constant, permettant de réduire la pente de descente absolue de manière linéaire,
3. une phase de descente parcourue avec une pente faible constante  $\gamma_2$  (et avec une vitesse verticale relativement faible), jusqu'à ce que l'arrondi final soit déclenché un peu avant le toucher des roues.

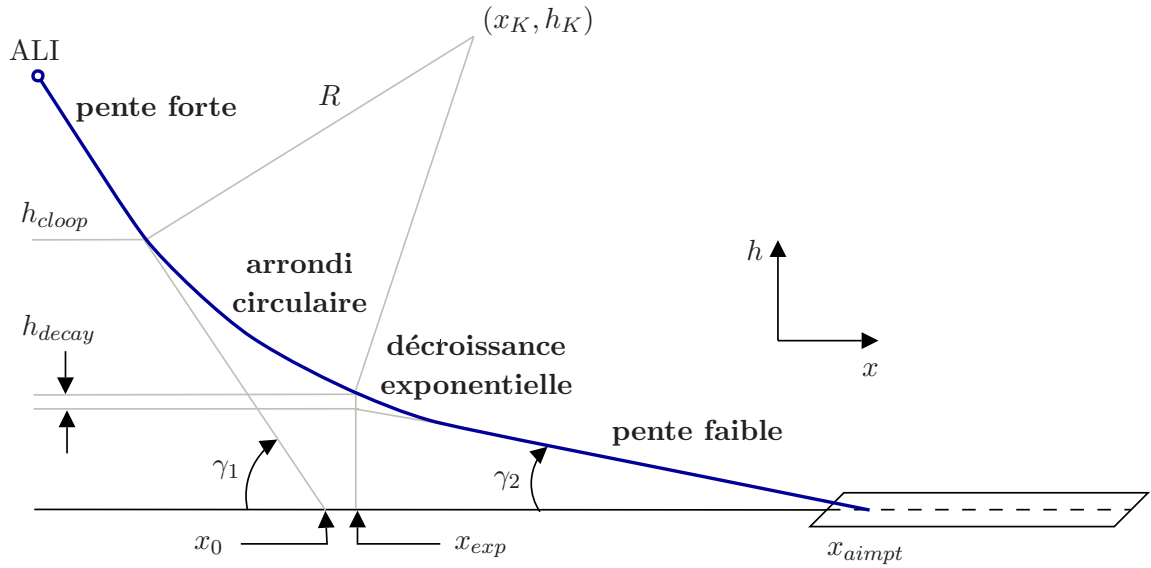


FIG. 1.1 – illustration de la phase d'atterrissage automatique.

**Remarque 1.5.** Si le véhicule passait directement de la phase d'arrondi circulaire à la phase de descente à pente faible, une discontinuité d'accélération pourrait apparaître. Par conséquent, une phase de décroissance exponentielle est utilisée afin de créer une transition douce entre ces deux segments (voir figure 1.1).

L'ensemble des paramètres permettant de décrire complètement la géométrie de la trajectoire d'atterrissage est regroupé dans le tableau 1.1. La description mathématique de

No.	Symbole	Paramètre
(1)	$\gamma_1$	pente forte
(2)	$\gamma_2$	pente faible
(3)	$x_{aimpt}$	intersection du segment de trajectoire à pente faible avec le sol
(4)	$h_{decay}$	altitude au début de la phase d'arrondi exponentiel
(5)	$R$	rayon de courbure de l'arrondi circulaire
(6)	$x_0$	intersection du segment de trajectoire à pente forte avec le sol
(7)	$x_K$	portée longitudinale du centre de l'arc de cercle de l'arrondi
(8)	$h_K$	altitude du centre de l'arc de cercle de l'arrondi circulaire
(9)	$x_{exp}$	portée longitudinale au début de l'arrondi exponentiel
(10)	$h_{clloop}$	altitude au début de l'arrondi circulaire
(11)	$\sigma$	taux de décroissance de l'arrondi exponentiel

TAB. 1.1 – paramètres de dimensionnement de la trajectoire d'atterrissage.

la géométrie de la trajectoire est donnée par

$$\begin{aligned} h_{steep} &= (x - x_0) \tan \gamma_1 && \text{si } h > h_{loop}, \\ h_{circ} &= h_K - \sqrt{R^2 - (x - x_K)^2} && \text{si } x > x_{exp}, \\ h_{exp} &= (x - x_{aimpt}) \tan \gamma_2 + h_{decay} \exp\left(\frac{x_{exp} - x}{\sigma}\right) && \text{si } h > 0, \end{aligned}$$

où  $h_{steep}$ ,  $h_{circ}$  et  $h_{exp}$  représentent l'altitude du véhicule durant chacune des phases de vol, pour une portée longitudinale donnée<sup>22</sup>  $x$ .

Le problème de planification de trajectoires d'atterrissage consiste alors à déterminer les valeurs des paramètres de dimensionnement de la trajectoire, de telle sorte que le véhicule se pose avec une pression dynamique donnée<sup>23</sup>, et que les équations dynamiques (1.64)-(1.67) soient vérifiées.

Les paramètres géométriques définis dans le tableau 1.1 peuvent être séparés en trois catégories distinctes :

- les paramètres 1 à 5 sont calculés une fois pour toutes, et ne requièrent aucune optimisation.
- Le paramètre 6,  $x_0$ , est utilisé de manière à satisfaire la contrainte sur la pression dynamique lors du toucher des roues de l'appareil sur la piste,
- Les paramètres 7 à 11 dépendent de  $x_0$ , et sont calculés de manière analytique.

Le diagramme 1.2 représente le processus global d'optimisation de trajectoires d'atterrissage proposé. Dans ce processus, certaines variables sont calculées ou fixées une fois pour toutes. Les valeurs des variables  $\gamma_1$ ,  $\gamma_2$ ,  $x_{aimpt}$ ,  $h_{decay}$  et  $R$  sont obtenues à partir de considérations sur la robustesse de la trajectoire aux incertitudes, le facteur de charge, et le taux de descente vertical au niveau du point de toucher. Ces variables n'étant pas directement liées à la contrainte sur la pression dynamique finale, elles ne nécessitent pas d'être itérées.

Ensuite, une valeur initiale de  $x_0$  est choisie (voir figure 1.2). Cette variable est le principal paramètre de contrôle du problème d'optimisation, et elle est utilisée afin de satisfaire la contrainte de pression dynamique au point de toucher. Les variables  $(x_K, h_K, x_{exp}, h_{loop}, \sigma)$  sont alors obtenues en imposant des contraintes de continuité sur la trajectoire. Elles dépendent toutes du paramètre  $x_0$ , et doivent être recalculées à chaque fois que  $x_0$  est modifié.

Maintenant, il est nécessaire de déterminer la valeur de la pression dynamique finale pour une géométrie donnée, et pour une condition initiale de pression dynamique (ou de vitesse relative) donnée. La détermination d'une solution analytique à ce problème n'est pas souhaitable, du fait du caractère fortement non linéaire du modèle aérodynamique du véhicule. Une propagation numérique du modèle (1.64)-(1.67), contraint à une géométrie de trajectoire donnée, est alors réalisée afin de déterminer la valeur finale de la contrainte de pression dynamique.

---

<sup>22</sup>par convention, la portée longitudinale  $x$  est choisie nulle au niveau du seuil de piste.

<sup>23</sup>ou, de manière équivalente, avec une vitesse relative donnée.

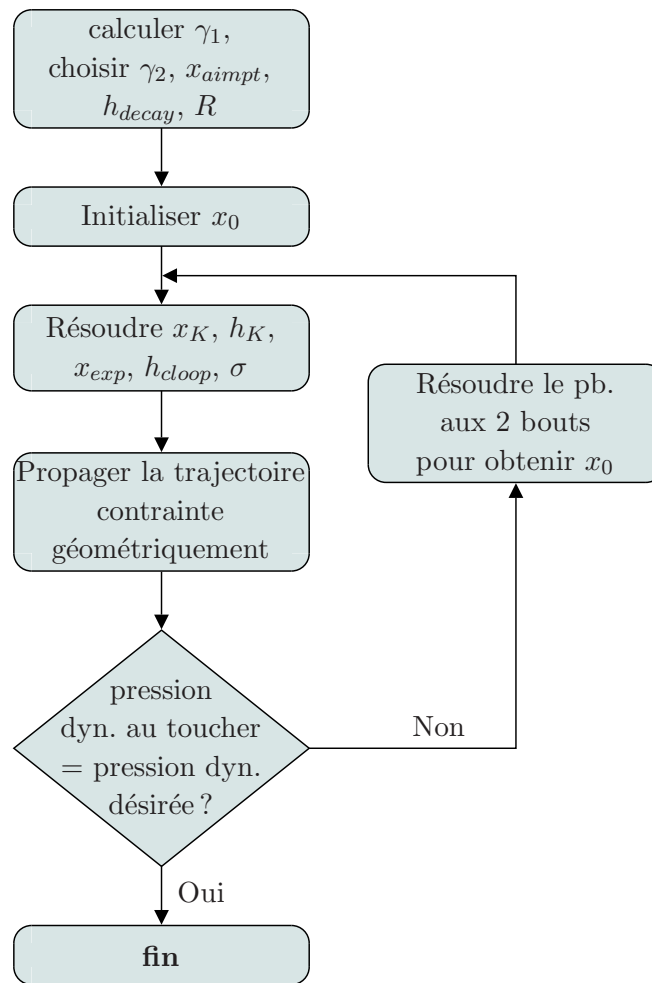


FIG. 1.2 – processus de planification de trajectoires d’atterrissage.

La valeur de  $x_0$  est alors modifiée jusqu’à ce que la pression dynamique finale soit dans un voisinage acceptable de la valeur désirée. Cela constitue un problème au deux bouts, pour lequel les valeurs initiale et finale de la pression dynamique sont connues. Ce problème est alors résolu par une méthode directe de tirs : pour chaque estimé de  $x_0$ , une trajectoire est propagée, fournissant ainsi la pression dynamique et le profil de la commande  $\alpha$  le long de la trajectoire candidate liée à  $x_0$ . Une technique de recherche par dichotomie donne alors la valeur optimale de  $x_0$  assurant le respect de la contrainte de pression dynamique au point de toucher.

La variation de  $x_0$  modifie essentiellement l’abscisse curviligne parcourue sur la trajectoire durant l’atterrissage automatique. La diminution de  $x_0$  diminue la longueur de la trajectoire, tel qu’illustré sur la figure 1.3. Ceci conduit à une réduction de l’énergie dissipée, associée à un temps de vol plus court, de telle sorte que le véhicule se pose avec une pression dynamique plus forte.



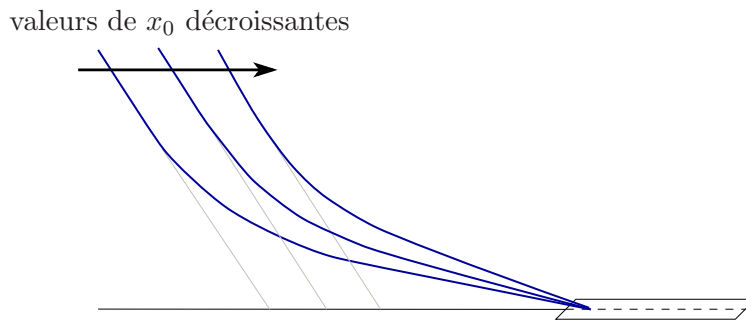


FIG. 1.3 – effet de la variation de  $x_0$  sur la trajectoire d'atterrissage.

## 1.5.2 Méthodes indirectes de tirs

### Description de la méthode

Considérons un problème de commande optimale simple, comportant une seule phase, et ne possédant pas de contraintes de trajectoire. Dans le cadre des méthodes indirectes de tirs, les variables d'optimisation sont choisies comme étant un sous-ensemble des variables intervenant dans les conditions d'optimalité issues du principe du maximum de Pontryagin. Dans le cas présent, on choisit les variables du problème NLP associé sous la forme

$$x = \{\lambda(t_0), t_f\},$$

les contraintes du problème NLP étant alors définies par

$$c(x) = \begin{pmatrix} \psi(x(t), p, t) \\ \lambda(t) - \Phi_x^T \\ (\Phi_t + H(x)) \end{pmatrix} \Big|_{t=t_f}, \quad (1.70)$$

où  $\psi(x(t), p, t)$  désigne les contraintes de bord (1.34) et (1.35) du problème de commande optimale. La principale différence entre les méthodes de tirs directs et indirects vient de la paramétrisation des variables de commande  $u(t)$ . Dans le premier cas, celles-ci sont décrites de manière implicite ou explicite par le biais d'un nombre fini de paramètres  $p$ . Dans le cas des méthodes indirectes de tirs, la commande est déterminée en chaque point par les équations (1.46) ou (1.42) associées au principe du maximum de Pontryagin. Les variables d'optimisation du problème NLP sont alors les états adjoints  $\lambda(t_0)$ , évalués à l'instant initial. Lorsque l'application directe du principe du maximum permet d'obtenir une expression explicite suffisamment simple des commandes, alors la propagation de la trajectoire (c'est à dire des ODE), peut être réalisée en utilisant une méthode aux valeurs initiales. Dans le cas contraire, lorsque les commandes sont obtenues sous forme implicite, il est nécessaire d'utiliser une méthode aux valeurs initiales pour intégrer numériquement les équations algébro-différentielles. Notons enfin que des contraintes inégalitaires de trajectoire (1.36) peuvent être prises en compte en les traitant comme une séquence d'arcs contraints et d'arcs non contraints (voir section 1.3.3).

En résumé, la méthode indirecte de tirs peut être résumée par l'algorithme 5. Comme

**Algorithme 5** : Méthode indirecte de tirs.

**Entrées** : Variables de décision  $x$ .

**Sorties** : Contraintes  $c(x)$  définies par (1.70).

**Algorithme** :

1 **tant que**  $c(x) \neq 0$  **faire**

2     **pour**  $k = 1, \dots, N$  **faire**

3         Initialiser la phase  $k$  :  $\{x^{(k)}(t_0^{(k)}), \lambda^{(k)}(t_0^{(k)}), p^{(k)}, t_0^{(k)}\}$ .

4         Résoudre le problème aux valeurs initiales : calculer  $(x^{(k)}(t_f^{(k)}), \lambda^{(k)}(t_f^{(k)}))$  en propageant la dynamique  $\dot{x} = f(x(t), t)$  de  $t_0^{(k)}$  à  $t_f^{(k)}$ .

5         Évaluer les contraintes de bord

$$c^{(k)}(x_f^{(k)}) = \begin{pmatrix} \psi(x^{(k)}(t_f^{(k)}), p^{(k)}, t_f^{(k)}) \\ \lambda(t_f^{(k)}) - \Phi_x^T|_{x=x^{(k)}(t_f^{(k)})} \\ (\Phi_t|_{t=t_f^{(k)}} + H(x)|_{x=x^{(k)}(t_f^{(k)})}) \end{pmatrix}.$$

6     **fin**

7     Ajuster les conditions initiales  $\{x^{(k)}(t_0^{(k)}), \lambda^{(k)}(t_0^{(k)}), p^{(k)}, t_0^{(k)}\}$ ,  $k = 1, \dots, N$ , à l'aide d'une méthode d'optimisation de type NLP, de manière que  $c^{(k)}(x_f^{(k)}) = 0$ ,  $k = 1, \dots, N$ .

8 **fin**

9 **retourner**  $c(x)$ .

la méthode directe de tirs, la méthode indirecte est particulièrement sensible à de faibles variations des conditions de bord. De ce fait, l'estimation initiale des paramètres d'optimisation doit être suffisamment proche de la solution optimale pour que l'algorithme 5 converge vers une solution admissible. Encore une fois, les techniques de tirs multiples, décrites dans la section suivante, permettront de réduire cette sensibilité numérique.

### Applications à l'optimisation de trajectoires de rentrée atmosphérique

Les techniques de tirs indirects ont donné lieu à un faible nombre de travaux dans le domaine de la rentrée atmosphérique. A titre d'illustration, nous décrivons ici une méthode de replanification de trajectoires à bord pour le véhicule expérimental X-37, sujet à des défauts actionneurs (blocage de l'aérofrein et/ou du body flap) lors des phases de dissipation d'énergie en zone terminale (TAEM), et d'atterrissage automatique (A&L) [36]. Sous l'hypothèse "Terre plate", la dynamique du centre de masse du véhicule peut être décrite par les équations suivantes, dans lesquelles la variable temporelle est remplacée par l'énergie totale définie par  $E = \frac{1}{2}mV^2 + mgh$ . On réalise donc le changement de variable

$$(\cdot)' = \frac{d(\cdot)}{dE} = \frac{1}{\dot{E}} \frac{d(\cdot)}{dt},$$

où la dérivée de l'énergie totale est donnée par

$$\frac{dE}{dt} = mV\dot{V} + mg\dot{h} = -DV.$$

On obtient alors :

$$x' = -\frac{1}{D} \cos \gamma \cos \chi, \quad (1.71)$$

$$y' = -\frac{1}{D} \cos \gamma \sin \chi, \quad (1.72)$$

$$h' = -\frac{1}{D} \sin \gamma, \quad (1.73)$$

$$V' = \frac{1}{mV} + \frac{g}{DV} \sin \gamma, \quad (1.74)$$

$$\gamma' = -\frac{L \cos \mu}{mDV^2} + \frac{g}{DV^2} \cos \gamma, \quad (1.75)$$

$$\chi' = -\frac{L \sin \mu}{mDV \cos \gamma}, \quad (1.76)$$

où  $x'$ ,  $y'$ ,  $h'$ ,  $V'$ ,  $\gamma'$  et  $\chi'$  représentent respectivement la portée longitudinale, la portée latérale, l'altitude, la vitesse relative, la pente et le cap, exprimés en fonction de l'énergie totale.

Les forces de portance  $L$  et de traînée  $D$  sont typiquement données par

$$L = \bar{q}SC_L, \quad (1.77)$$

$$D = \bar{q}SC_D, \quad (1.78)$$

où  $\bar{q} = \frac{1}{2}\rho V^2$  est la pression dynamique, et  $C_L$ ,  $C_D$  représentent respectivement les coefficients de portance et de traînée tels que

$$C_L(\alpha, M, \lambda_{CL}) = CL_\alpha \alpha + CL_{(\alpha, M)} \alpha M + CL_M M + (1 - \lambda_{CL}) \theta_{\underline{CL}} + \lambda_{CL} \theta_{\overline{CL}}, \quad (1.79)$$

$$C_D(\alpha, M, \lambda_{CD}) = CD_{\alpha\alpha} \alpha^2 + CD_\alpha \alpha + CD_{(\alpha, M)} \alpha M + CD_M M + (1 - \lambda_{CD}) \theta_{\underline{CD}} + \lambda_{CD} \theta_{\overline{CD}}. \quad (1.80)$$

Les paramètres supplémentaires  $\lambda_{CL}$  et  $\lambda_{CD}$  permettent alors de choisir un modèle aérodynamique valide en fonction du défaut actionneur rencontré, tel que

$$0 \leq \lambda_{CL} \leq 1, \quad (1.81)$$

$$0 \leq \lambda_{CD} \leq 1, \quad (1.82)$$

$$|\lambda_{CL} - \lambda_{CD}| \leq \lambda_r, \quad (1.83)$$

où  $\lambda_r$  est choisi de telle sorte que le modèle aérodynamique soit atteignable<sup>24</sup>. Les paramètres  $\theta_{\underline{CL}}$ ,  $\theta_{\overline{CL}}$ ,  $\theta_{\underline{CD}}$  et  $\theta_{\overline{CD}}$  permettent de représenter les domaines de conditions

---

<sup>24</sup>par exemple, pour éviter d'utiliser le modèle associé à la portance maximale avec le modèle associé à la traînée minimale.

d'équilibre atteignables en fonctionnement défaillant. En situation nominale, on a

$$\begin{aligned} \theta_{\underline{CL}} &= CL_{0_{\min}}, & \theta_{\overline{CL}} &= CL_{0_{\max}}, \\ \theta_{\underline{CD}} &= CD_{0_{\min}}, & \theta_{\overline{CD}} &= CD_{0_{\max}}. \end{aligned} \quad (1.84)$$

La phase de dissipation d'énergie en zone terminale (TAEM) débute lorsque le véhicule atteint Mach 2, à environ 25 km d'altitude, et se termine à Mach 0.5, à environ 5 km d'altitude (voir figure 1.4(a)). L'objectif de la loi de guidage durant cette phase est double :

1. dissiper l'énergie du véhicule afin d'assurer un atterrissage fiable,
2. aligner le véhicule avec l'axe de piste.

La phase d'atterrissage automatique (A&L) consiste alors à satisfaire les contraintes de vitesse verticale et de pression dynamique au niveau du point de toucher des roues sur la piste (voir figure 1.4(b)). Afin de développer l'algorithme de planification de trajectoires

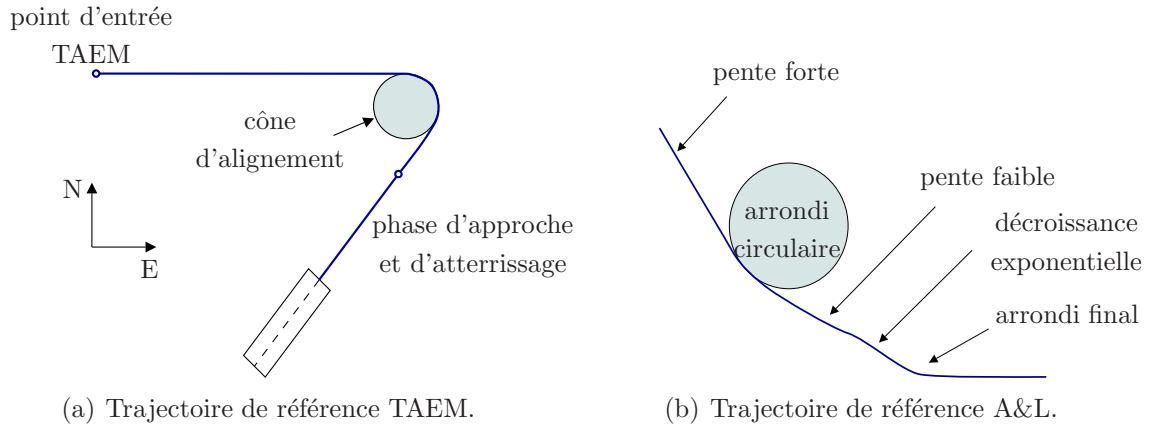


FIG. 1.4 – description géométrique des trajectoires TAEM et A&L de référence.

TAEM/A&L, les trajectoires sont décrites en fonction de quatre paramètres géométriques qui peuvent être modifiés par l'algorithme de replanification de trajectoires à bord :

- deux paramètres géométriques longitudinaux, qui permettent de modifier les profils d'altitude, de vitesse et de pente sur la trajectoire : l'altitude  $h_{ff}$  à laquelle débute la manoeuvre d'arrondi final, et la pression dynamique  $\bar{q}_{TD}$  au niveau du toucher des roues de l'appareil sur la piste,
- deux paramètres géométriques latéraux, permettant de modifier la trace au sol<sup>25</sup> : le cap initial  $\chi_0$  au point d'entrée TAEM, et l'altitude  $h_{HAC}$  à laquelle débute l'enroulement autour du cône d'alignement (HAC).

En rappelant que l'enveloppe aérodynamique du véhicule à l'équilibre est déterminée en ajustant les termes  $\lambda_{CL}$  et  $\lambda_{CD}$  dans les expressions (1.79)-(1.80), le vecteur de paramètres

<sup>25</sup>on appelle typiquement *trace au sol* la projection de la trajectoire de vol sur un plan horizontal lié au sol.

permettant de décrire complètement la trajectoire TAEM/A&L est donné par

$$d = \begin{pmatrix} \chi_0 \\ h_{HAC} \\ h_{ff} \\ \bar{q}_{TD} \\ \lambda_{CL} \\ \lambda_{CD} \end{pmatrix}.$$

Une fois que ces paramètres sont fixés, une architecture de commande basée sur l'inversion dynamique de modèle est utilisée afin de générer les consignes de guidage, c'est à dire l'incidence  $\alpha$  et la gîte aérodynamique  $\mu$ , permettant de suivre les trajectoires de référence des états  $(x, y, h, V, \gamma, \chi)$  associées à la géométrie spécifiée.

La génération à bord des trajectoires TAEM/A&L consiste alors à déterminer le vecteur de paramètres  $d$  permettant d'annuler les erreurs finales

$$f_x(d) = x(E_{TD}) - x_{TD}, \quad (1.85)$$

$$f_y(d) = y(E_{TD}) - y_{TD}, \quad (1.86)$$

$$f_h(d) = h(E_{TD}) - h_{TD}, \quad (1.87)$$

$$f_{\dot{h}}(d) = \dot{h}(E_{TD}) - \dot{h}_{TD}, \quad (1.88)$$

$$(1.89)$$

où l'indice  $TD$  correspond à la valeur de l'état obtenue au point de toucher, tout en vérifiant la dynamique du véhicule décrite par les équations (1.71)-(1.76).

**Remarque 1.6.** Comme les équations (1.71)-(1.76) sont intégrées par rapport à l'énergie, il n'est pas nécessaire de rajouter une contrainte de vitesse finale  $V_{TD}$  : lorsque  $f_h$  est vérifiée, alors l'équation  $E = \frac{1}{2}mV^2 + mgh$  permet de déterminer la vitesse relative  $V_{TD}$  correspondante. De la même manière, aucune contrainte n'est imposée sur la pente finale de la trajectoire A&L : lorsque la vitesse finale  $V_{TD}$  est connue et l'équation  $f_{\dot{h}}$  satisfaite, alors l'équation dynamique  $\dot{h} = V \sin \gamma$  fournira automatiquement une pente  $\gamma_{TD}$  acceptable.

A chaque itération  $i$  de l'algorithme de replanification de trajectoires, le vecteur de paramètres  $d$  est déplacé dans une direction  $\zeta$  permettant de réduire les erreurs définies par les équations (1.85)-(1.88), telle que

$$d(i+1) = d(i) + K\zeta(i),$$

où  $K$  représente le gain de correction des paramètres dans la direction  $\zeta$ . Le vecteur de directions, recalculé à chaque itération, est alors solution du problème d'optimisation

$$\begin{aligned} \min_{\zeta \in \mathbb{R}^6} \quad & J = \frac{1}{2} \zeta^T W \zeta, & (1.90) \\ \text{t.q.} \quad & \text{a) } A\zeta = F, \\ & \text{b) } B\zeta \leq g, \end{aligned}$$

La première contrainte du problème (1.90) permet de garantir que le vecteur  $\zeta$  annule de manière locale les erreurs finales (1.85)-(1.88). La matrice  $A$  contient alors des dérivées partielles des fonctions (1.85)-(1.88) par rapport aux paramètres de la trajectoire, et le vecteur  $F$  contient les valeurs réelles de ces erreurs finales calculées à partir des fonctions non linéaires (1.85)-(1.88). la seconde contrainte de (1.90) permet de satisfaire les inégalités (1.81)-(1.83) portant sur la sélection d'un modèle aérodynamique valide en fonctionnement défaillant. Finalement, la matrice de pondération diagonale  $W$  dans la fonction de coût quadratique, permet de pénaliser le mouvement du véhicule dans la direction  $\chi_0$ , afin de minimiser l'amplitude des manoeuvres latérales.

Afin de résoudre le problème d'optimisation (1.90), les contraintes inégalitaires sont ignorées dans un premier temps. La solution du problème réduit, de type "pseudo-inverse pondérée", est alors donnée par

$$\zeta = W^{-1}A^T (AW^{-1}A^T)^{-1} F.$$

Les contraintes inégalitaires sont alors vérifiées et, si l'une d'entre elle est violée, on forme alors la matrice augmentée  $\hat{B}$  comprenant uniquement les lignes de  $B$  correspondant aux contraintes non respectées. Le vecteur  $\hat{g}$  est formé de la même manière à partir de  $g$ . La solution du nouveau problème est alors donnée par

$$\zeta = W^{-1} \left[ A^T (AW^{-1}A^T)^{-1} F + \left( \hat{B}^T - A^T (AW^{-1}A^T)^{-1} AW^{-1}\hat{B}^T \right) \eta \right],$$

où le multiplicateur de Lagrange  $\eta$  est donné par

$$\eta = \left[ \hat{B}W^{-1}\hat{B}^T - \hat{B}W^{-1}A^T (AW^{-1}A^T)^{-1} AW^{-1}\hat{B}^T \right]^{-1} \left( \hat{g} - \hat{B}W^{-1}A^T (AW^{-1}A^T)^{-1} F \right).$$

Les éléments de la matrice  $F$  sont construits en ligne, en propageant la trajectoire afin de prédire la valeur des commandes de référence au point de toucher. De plus, afin d'approximer les dérivées partielles dans la matrice  $A$ , chaque paramètre de trajectoire contenu dans le vecteur  $d$  doit être perturbé, et son impact sur l'état final prédit en intégrant numériquement la trajectoire entre  $E_0$  et  $E_{TD}$  (problème de perturbations singulières).

### 1.5.3 Méthodes de tirs multiples

#### Description de la méthode

Comme nous l'avons indiqué dans les deux sections précédentes, les méthodes de tirs simples sont relativement sensibles aux conditions initiales, ce qui entraîne souvent des écarts très importants à l'autre bout de la trajectoire. Afin de réduire cette sensibilité, les méthodes de tirs multiples consistent à réduire la portée des tirs, en partitionnant l'espace temps en une série d'intervalles plus courts. Ces techniques ont été introduites dans [55] pour pallier à certains inconvénients associés à certains problèmes aux deux bouts.

Sous sa forme la plus simple, le problème de tirs multiples consiste à déterminer la valeur initiale  $\nu(t_0) = \nu_0$  telle que la condition de bord

$$0 = \phi(\nu(t_f), t_f), \tag{1.91}$$

soit satisfaite pour une valeur  $t_0 < t_f$  vérifiant

$$\dot{\nu} = f(\nu(t), t). \quad (1.92)$$

Commençons par créer une partition du domaine temporel en  $M$  segments tels que

$$t_0 < t_1 < \dots < t_M = t_f.$$

Notons par  $x_j$ ,  $j = 0, \dots, M - 1$ , la valeur initiale des variables  $\nu$  au début de chaque segment, et par  $\bar{x}_j$  l'intégration de la dynamique (1.92) le long du segment  $j$ , entre les instants  $t_j$  et  $t_{j+1}$ .

Pour l'ensemble des segments, on obtient alors l'ensemble  $\nu$  des variables du problème NLP, tel que

$$x = \{\nu_0, \nu_1, \dots, \nu_{M-1}\}. \quad (1.93)$$

Il est également nécessaire d'introduire un ensemble de contraintes supplémentaires, appelées *contraintes de résidus*, permettant d'assurer que les différents segments se joignent à leurs extrémités :

$$c(x) = \begin{pmatrix} \nu_1 - \bar{\nu}_0 \\ \nu_2 - \bar{\nu}_1 \\ \vdots \\ \phi(\nu_M, t_f) \end{pmatrix}. \quad (1.94)$$

Ces contraintes de résidus engendrent une augmentation de la taille du problème NLP qui devra être résolu. Le nombre de variables et de contraintes pour la méthode de tirs multiples est  $n = n_\nu M$ , où  $n_\nu$  représente le nombre de variables dynamiques  $\nu$  du système (1.92), et  $M$  le nombre de segments. Cependant, la matrice jacobienne  $G$  associée à la direction de recherche de Newton (1.4) possède la propriété d'être creuse (seuls  $Mn_\nu^2$  éléments sont non nuls), et des algorithmes d'optimisation efficaces, tenant compte de manière explicite de cette propriété<sup>26</sup>, peuvent alors être utilisés [43].

Bien que la technique de tirs multiples permet d'améliorer de manière significative la robustesse des méthodes directes et indirectes de tirs simples, le nombre de variables NLP additionnelles et de contraintes de résidus associées rendent son implémentation algorithmique délicate. La prise en compte du caractère creux de la matrice est alors nécessaire afin de résoudre de manière efficace les équations de Newton associées au problème NLP.

Les principales étapes de la démarche associée à la méthode de tirs multiples sont décrites dans l'algorithme 6.

**Remarque 1.7.** La méthode de tirs multiples peut être formulée de manière directe ou indirecte, selon la définition de l'ensemble  $\nu$  des variables d'optimisation. Si  $\nu = (x, u)$  est choisi comme étant l'état et les commandes du système (1.92), alors la méthode sera directe. En revanche, si  $\nu$  comprend en plus les états adjoints, i.e.,  $\nu = (x, u, \lambda)$ , alors l'algorithme de tirs multiples sera de type indirect.

---

<sup>26</sup>communément appelée propriété de *sparsité* de la matrice.

**Algorithme 6** : Méthode de tirs multiples.

**Entrées** : Variables de décision  $x$ .

**Sorties** : Contraintes  $c(x)$  définies par (1.94).

**Algorithme** :

```

1 tant que  $c(x) \neq 0$  faire
2   pour  $k = 1, \dots, N$  faire
3     Initialiser la phase  $k$  :
4     pour  $j = 0, \dots, M - 1$  faire
5       Initialiser le segment  $j + 1$  :  $\{\nu_j^{(k)}, t_j^{(k)}\}$ .
6       Résoudre le problème aux valeurs initiales : calculer  $\bar{\nu}_j^{(k)}$  en propageant
          la dynamique  $\dot{x} = f(x(t), t)$  de  $t_j^{(k)}$  à  $t_{j+1}^{(k)}$ .
7       Évaluer les contraintes de bord  $\nu_{j+1}^{(k)} - \bar{\nu}_j^{(k)}$ .
8     fin
9     Calculer la contrainte  $\phi(\nu_M^{(k)}, t_f^{(k)})$ .
10  fin
11  Ajuster les conditions initiales  $x^{(k)} = \{\nu_0^{(k)}, \nu_1^{(k)}, \dots, \nu_{M-1}^{(k)}\}$ ,  $k = 1, \dots, N$ , à
          l'aide d'une méthode d'optimisation de type NLP, de manière que  $c(x^{(k)}) = 0$ ,
           $k = 1, \dots, N$ .
12 fin
13 retourner  $c(x)$ .
```

### Applications à l'optimisation de trajectoires de rentrée atmosphérique

L'application des techniques de tirs multiples dans le cadre de la rentrée atmosphérique a engendré un nombre important de travaux de recherche dans le domaine [56, 57, 58, 59] (voir le tableau 1.3 pour plus de références). A titre d'illustration, nous avons choisi de décrire une technique d'optimisation de trajectoires de dissipation d'énergie en zone terminale (TAEM) pour le véhicule expérimental X-33, développée dans [60]. Sous l'hypothèse "Terre plate", les équations à trois degrés de liberté du véhicule de rentrée atmosphérique sont données par [32]

$$\dot{x} = V \cos \gamma \cos \chi, \quad (1.95)$$

$$\dot{y} = V \cos \gamma \sin \chi, \quad (1.96)$$

$$\dot{h} = V \sin \gamma, \quad (1.97)$$

$$\dot{V} = -\frac{D}{m} - g \sin \gamma, \quad (1.98)$$

$$\dot{\gamma} = \frac{L \cos \mu}{mV} - \frac{g}{V} \cos \gamma, \quad (1.99)$$

$$\dot{\chi} = \frac{L \sin \mu}{mV \cos \gamma}, \quad (1.100)$$

où les forces de portance et de traînée  $L$  et  $D$  sont définies par (1.77) et (1.78), avec  $\bar{q} = \frac{1}{2}\rho V^2$ .



Afin de permettre la génération de trajectoires à bord, les coefficients aérodynamiques  $C_L(\alpha, M)$  et  $C_D(\alpha, M)$  sont approximés par des fonctions simples. Notamment, une équation polaire standard est utilisée pour approximer le coefficient de traînée, telle que

$$C_D(\alpha, M) = C_{D_0} + KC_L^N, \quad (1.101)$$

où les scalaires  $K$ ,  $N$ , et le coefficient de traînée à portance nulle  $C_{D_0}$ , sont obtenus pour différents nombres de Mach en effectuant une simple approximation aux moindres carrés.

La trajectoire TAEM de référence est paramétrée par un petit nombre de segments géométriques, représentés sur la figure 1.5. Ce sont essentiellement les mêmes segments de vol que ceux utilisés pour le guidage de la navette spatiale américaine :

- une phase d'acquisition du cône d'alignement (HAC), où le véhicule effectue un virage à rayon constant à partir du point d'entrée TAEM, afin de s'aligner avec un point tangent sur le HAC,
- une droite de ralliement parcourue à ailes horizontales jusqu'au point de tangence avec le cône d'alignement,
- une phase d'enroulement spiral autour du HAC, permettant d'aligner le véhicule avec l'axe de piste,
- une phase d'approche finale, où le véhicule vole à ailes droites jusqu'à atteindre l'interface avec la phase d'atterrissage automatique A&L (point ALI).

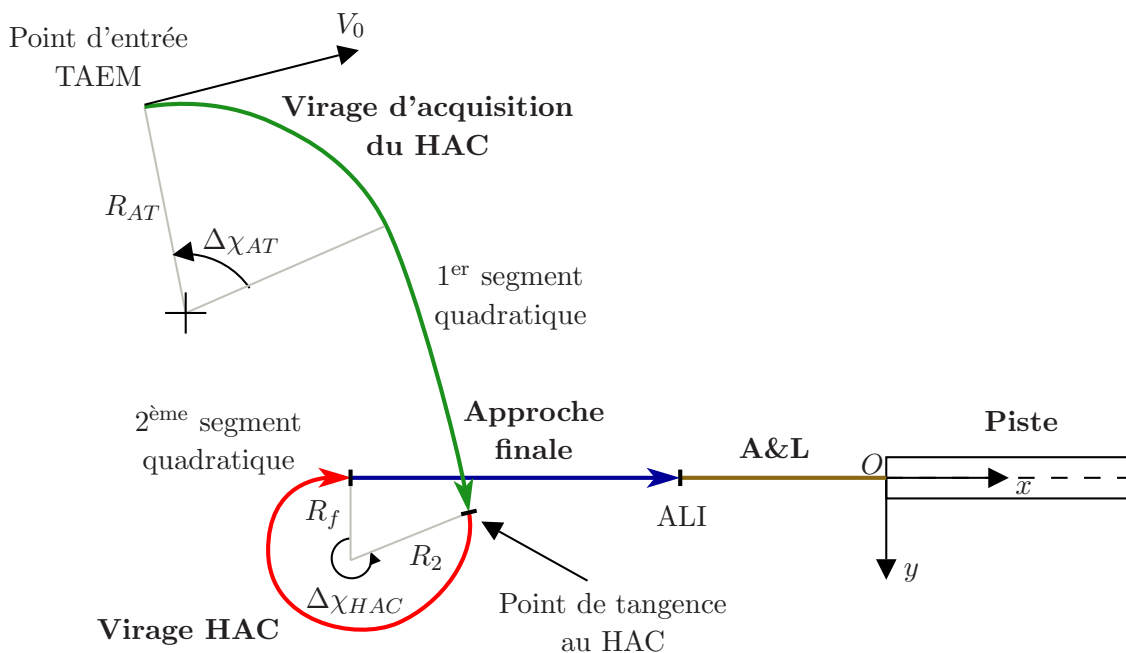


FIG. 1.5 – segmentation de la trajectoire de référence TAEM.

Une partie essentielle de la méthode de génération de trajectoires TAEM proposée dans [60] consiste à prédire de manière précise la longueur restante de la trace au sol jusqu'au point ALI. Cette distance, appelée classiquement “range-to-go” ( $s_{togo}$ ) au début de la

phase TAEM, comprend la contribution de chaque segment de vol et est donnée par

$$s_{togo} = s_{AT} + s_{HAC} + s_{PF},$$

où  $s_{AT}$ ,  $s_{HAC}$  et  $s_{PF}$  correspondent aux longueurs restantes prédites de la trace au sol pour les phases d'acquisition, d'enroulement spiral autour du HAC, ainsi que pour la phase d'approche finale. Une expression analytique de la longueur au sol pour la phase d'acquisition du HAC est donnée par

$$s_{AT} = R_{AT}|\Delta\chi_{AT}|, \quad (1.102)$$

où  $\Delta\chi_{AT}$  représente l'erreur initiale de cap au niveau du virage d'acquisition (voir figure 1.5), et  $R_{AT}$  représente le rayon approché du virage d'acquisition tel que

$$R_{AT} = \frac{V^2 \cos \gamma}{g \tan \mu_{AT}}, \quad (1.103)$$

où  $\mu_{AT}$  est un angle de gîte nominal (constant) prédit. Le rayon de virage du HAC est une fonction quadratique de l'angle de virage  $\Delta\chi_{HAC}$  (voir figure 1.5), telle que

$$R_{HAC} = R_f + R_2\Delta\chi_{HAC}^2, \quad (1.104)$$

où  $R_2$  désigne le rayon de virage initial et  $R_f$  le rayon de virage final. L'intégration de (1.104) par rapport à l'angle de virage du HAC  $\Delta\chi_{HAC}$  fournit alors la longueur de la trace au sol pour la phase d'enroulement spiral autour du HAC, tel que

$$s_{HAC} = R_f\Delta\chi_{HAC} + \frac{1}{3}R_2\Delta\chi_{HAC}^3. \quad (1.105)$$

Finalement, la longueur au sol associée à la phase d'approche finale est simplement donnée par

$$s_{PF} = x_{ALI} - x_{HAC}. \quad (1.106)$$

En résumé, la longueur de la trace au sol pour la phase de dissipation d'énergie en zone terminale peut être calculée grâce aux équations (1.102)-(1.106), en fonction des paramètres géométriques  $\mu_{AT}$ ,  $R_f$ ,  $R_2$ ,  $x_{ALI}$  et  $x_{HAC}$ .

Le profil d'altitude le long de la phase TAEM est défini par deux polynômes quadratiques de la forme

$$h_{ref} = a_0 + a_1s + a_2s^2, \quad (1.107)$$

où  $s$  représente l'abscisse curviligne au sol. Le premier polynôme quadratique débute au point d'entrée TAEM et se termine au point de tangence avec le HAC, et le second polynôme quadratique commence au niveau du virage HAC et se termine au point de sortie ALI (voir figure 1.5). Une partie des coefficients de ces polynômes est directement calculée à partir de considérations géométriques sur la trajectoire TAEM. Le tableau 1.2 regroupe les paramètres de dimensionnement associés au profil d'altitude considéré.

1 <sup>er</sup> polynôme	2 <sup>ème</sup> polynôme
$a_0 = h_0$	$a_0 = h_{HAC}$
$a_1 = \tan \gamma_0$	$a_1 = \tan \gamma_{HAC}$
$a_2 = \frac{\tan \gamma_{HAC} - \tan \gamma_0}{2s_{AT}}$	$a_2 = \frac{\tan \gamma_{ALI} - \tan \gamma_{HAC}}{2(s_{HAC} + s_{PF})}$

TAB. 1.2 – paramètres de dimensionnement des polynômes quadratiques de référence.

**Remarque 1.8.** Pour la majorité de la phase TAEM, l'algorithme de planification de trajectoires de la navette spatiale américaine utilise un polynôme cubique afin de paramétrer l'altitude de référence. De plus, une trajectoire de descente de référence à pente constante, relativement courte, est utilisée au début de la phase TAEM si le véhicule se trouve au-delà d'une valeur de trace au sol prédéfinie.

Une fois que la géométrie de la trajectoire TAEM est complètement spécifiée, les expressions analytiques précédentes peuvent être manipulées et inversées, afin de déterminer les commandes de guidage de référence, i.e., l'angle de gîte aérodynamique  $\mu$  et le facteur de charge normal  $n_z$ .

Le processus de planification de trajectoires TAEM comporte quatre étapes, décrites dans l'algorithme 7. Les trajectoires TAEM admissibles sont déterminées en itérant sur un pe-

<b>Algorithme 7</b> : Planification de trajectoires TAEM.
<b>Entrées</b> : Paramètres de dimensionnement géométrique de la trajectoire.
<b>Sorties</b> : Trajectoire TAEM optimale.
<b>Algorithme</b> :
1 Prédiction de la longueur de la trace au sol.
2 Propagation de la trajectoire du point d'entrée TAEM jusqu'au point de tangence au HAC.
3 Propagation de la trajectoire pour les phases d'enroulement spiral autour du HAC et d'approche finale jusqu'au point ALI.
4 Sélection de la trajectoire de référence optimale parmi l'ensemble de trajectoires admissibles obtenues.

tit nombre de paramètres géométriques ( $R_f$ ,  $R_2$  et  $x_{HAC}$ , les autres étant ensuite déduits des précédents de manière analytique), et en imposant des contraintes appropriées sur la dynamique (1.95)-(1.100) durant la propagation des trajectoires. Un classement de l'ensemble fini de trajectoires admissibles obtenues en fonction d'un critère de performance donné fournit alors la trajectoire de référence optimale.

**Remarque 1.9.** La trajectoire TAEM étant ici propagée sur plusieurs segments, la méthode d'optimisation utilisée correspond bien à une méthode de tirs multiples.

La propagation de la trajectoire TAEM jusqu'au point de tangence avec le cône d'alignement est réalisée en propageant la "hauteur d'énergie<sup>27</sup>"  $E = \frac{V^2}{2g} + h$ , en fonction de

<sup>27</sup>exprimée en unité de distance.

l'abscisse curviligne au sol  $s$ , telle que

$$\frac{dE}{ds} = -\frac{D}{mg \cos \gamma_{ref}}.$$

Un schéma d'intégration simple, de type Euler, est alors utilisé pour propager la hauteur d'énergie à partir de  $E_0$  :

$$E_{k+1} = E_k + \left. \frac{dE}{ds} \right|_k \Delta s,$$

où  $\Delta s = s_{ac}/N$ , et  $N$  représente le nombre total de pas d'intégration. Les profils de référence des états  $x$ ,  $y$ ,  $h$ ,  $V$ ,  $\gamma$  et  $\chi$  sont alors obtenus de manière analytique le long de la trajectoire de référence.

La seconde propagation de trajectoire, réalisée à partir du virage spiral jusqu'à la fin du segment rectiligne d'approche finale, est réalisée de manière similaire à la première propagation. La principale différence est que la variable indépendante d'intégration considérée est l'énergie, de telle manière que la propagation se termine à un niveau d'énergie donné au point ALI. Par conséquent, l'équation de propagation avec le schéma simple d'intégration de type Euler, est donnée par

$$s_{k+1} = s_k + \left. \frac{ds}{dE} \right|_k \Delta E, \quad (1.108)$$

où  $\Delta E = (E_{ALI} - E_{HAC})/N$ , et  $N$  désigne le nombre total de pas d'intégration. La faisabilité de la trajectoire TAEM obtenue peut alors être testée en calculant l'erreur d'altitude

$$|h_{ALI} - h_{end}| < \varepsilon,$$

où  $h_{end}$  représente l'altitude finale de la trajectoire propagée, obtenue à partir de l'expression (1.107) et de la valeur finale de la distance au sol définie par (1.108), et  $\varepsilon$  est une tolérance donnée.

Une fois qu'une trajectoire TAEM admissible est obtenue, les paramètres géométriques de dimensionnement sont mémorisés, et l'algorithme de propagation est relancé en choisissant un autre jeu de paramètres  $x_{HAC}$ ,  $R_f$  et  $R_2$ . Lorsque toutes les itérations sont réalisées, les trajectoires admissibles sont triées en fonction d'un indice de performance donné, et la meilleure configuration géométrique est alors conservée.

## 1.5.4 Méthodes de transcription indirecte

### Description de la méthode

Comme nous l'avons indiqué en préambule de la section 1.3, le problème de commande optimale peut être considéré comme un problème de dimension infinie, les variables à optimiser étant le plus souvent de nature continue. Il est alors nécessaire de transcrire ou de convertir le problème de dimension infinie en un problème de dimension finie : c'est le principe des *techniques de transcription*, aussi appelées *méthodes de collocation*, que

nous allons détailler dans cette section. Ces méthodes ont été initialement développées pour résoudre les problèmes aux deux bouts. Une méthode de transcription comprend principalement trois étapes :

1. convertir le système dynamique en un problème comprenant un nombre fini de variables,
2. résoudre le problème de dimension finie grâce à une méthode d'optimisation paramétrique, par exemple de type NLP,
3. analyser la qualité de la solution obtenue et, si nécessaire, répéter les étapes de transcription et d'optimisation.

La première étape revient à partitionner l'espace-temps en un nombre fini de segments, c'est à dire

$$t_0 < t_1 < \dots < t_M = t_f.$$

Supposons que nous ayons à résoudre le problème aux deux bouts simple décrit par (1.91) et (1.92). Considérons une seule phase ( $k = 1$ ) et analysons la méthode de tirs multiples, résumée dans l'algorithme 6. Pour le segment  $j$ , il est nécessaire de satisfaire la *contrainte de résidu*

$$\nu_{j+1} - \bar{\nu}_j. \tag{1.109}$$

Supposons que l'estimé  $\bar{\nu}_j$  soit obtenu par la méthode d'intégration d'Euler explicite (1.55). Nous obtenons alors

$$\nu_{j+1} - (\nu_j + h_j f_j),$$

où  $t_{j+1} = t_j + h_j$  pour chaque segment  $j = 1, \dots, M - 1$ . Bien sûr, l'utilisation d'une méthode d'intégration de type Runge-Kutta ou trapézoïdale est également possible. Un choix très répandu dans le cadre des méthodes de collocation est celui de l'intégration de type Hermite-Simpson. L'expression (1.109) devient alors

$$0 = \nu_{j+1} - \nu_j - \frac{h_{j+1}}{6}(f_{j+1} + 4\bar{f}_{j+1} + f_j) = \zeta_j. \tag{1.110}$$

Le terme  $\zeta_j$  est alors appelé *résidu de Simpson*, et l'interpolant de Hermite est alors défini par

$$\bar{\nu}_{j+1} + \frac{1}{2}(\nu_j + \nu_{j+1}) + \frac{h_{j+1}}{8}(f_j - f_{j+1}),$$

pour les variables situées au milieu de l'intervalle.

La méthode de transcription indirecte est donc similaire à celle de tirs multiples, la principale différence étant que la première utilise un schéma d'intégration implicite, alors que la seconde se focalise sur la résolution d'un problème aux valeurs initiales, c'est à dire de propagation de trajectoires. Dans le cas de la méthode de collocation indirecte, l'optimisation porte sur les variables du système ainsi que sur les états adjoints, i.e.,  $\nu = (x, u, \lambda)$ . Ceci impose donc que le calcul des équations adjointes (1.41) soit rendu possible. La solution du problème de collocation est un polynôme continu par morceaux, vérifiant la dynamique du système et les contraintes en un ensemble fini de points, appelés *points de*

collocation, dans l'intervalle  $t_j < t < t_{j+1}$ .

L'algorithme 8 regroupe les principales étapes de la démarche.

<b>Algorithme 8</b> : Méthode de transcription indirecte.	
<b>Entrées</b> : Variables de décision $x$ définies par (1.93).	
<b>Sorties</b> : Contraintes de résidus $c(x)$ .	
<b>Algorithme</b> :	
1	<b>tant que</b> $\zeta \neq 0$ <b>faire</b>
2	<b>pour</b> $k = 1, \dots, N$ <b>faire</b>
3	Initialiser la phase $k$ :
4	<b>pour</b> $j = 0, \dots, M - 1$ <b>faire</b>
5	Évaluer la contrainte de résidu $\zeta_j^{(k)}$ , par exemple avec
	$\zeta_j^{(k)} = \nu_{j+1}^{(k)} - \nu_j^{(k)} - \frac{h_{j+1}}{6}(f_{j+1} + 4\bar{f}_{j+1} + f_j).$
6	<b>fin</b>
7	Calculer la contrainte $\phi(\nu_M^{(k)}, t_f^{(k)})$ dans (1.94).
8	<b>fin</b>
9	Ajuster les variables d'optimisation $x^{(k)} = \{\nu_0^{(k)}, \nu_1^{(k)}, \dots, \nu_{M-1}^{(k)}\}$ , $k = 1, \dots, N$ , à l'aide d'une méthode d'optimisation de type NLP, de manière que $\zeta^{(k)} = 0$ , $k = 1, \dots, N$ .
10	<b>fin</b>
11	<b>retourner</b> $\zeta$ .

**Remarque 1.10.** Les contraintes inégalitaires de trajectoire du type (1.36) peuvent être prises en compte dans l'algorithme d'optimisation 8, en les traitant comme une séquence d'arcs contraints et d'arcs non contraints (voir section 1.3.3).

### Applications à l'optimisation de trajectoires de rentrée atmosphérique

A l'heure actuelle, peu de travaux ont été effectués dans le cadre de l'optimisation de trajectoires de rentrée atmosphérique par collocation indirecte. On peut toutefois citer l'article [61], dans lequel un algorithme d'optimisation hybride est proposé afin d'optimiser des trajectoires de secours (trajectoires "abort") pour le X-33, générées à bord du véhicule lors de l'apparition d'une situation défaillante pouvant compromettre la mission. Les conditions d'optimalité du problème d'optimisation sont alors déterminées de manière explicite, et le vecteur de variables de collocation est choisi de façon à contenir les variables du système et les états adjoints.

### 1.5.5 Méthodes de transcription directe

#### Description de la méthode

La méthode de transcription directe, ou de collocation directe, est sans nul doute la technique d'optimisation de trajectoires la plus répandue dans la communauté aéronautique et spatiale. Cette méthode présente deux avantages considérables :

- comme pour les autres méthodes d'optimisation de type direct, elle ne repose pas explicitement sur les conditions d'optimalité du principe du maximum de Pontryagin,
- contrairement aux autres techniques, elle ne nécessite pas d'établir un séquençement des arcs contraints des arcs non contraints lorsque des contraintes inégalitaires de trajectoire sont imposées.

De ce fait, la méthode de transcription directe permet de considérer des applications relativement complexes en assurant un niveau de versatilité et de robustesse important, en comparaison avec les méthodes d'optimisation de type indirect ou les techniques de tirs simples/multiples.

Comme pour la méthode de collocation indirecte, la première étape consiste à partitionner le domaine temporel en  $M$  segments entre les instants  $t_0$  et  $t_f$ , tel que

$$t_0 < t_1 < \dots < t_M = t_f.$$

Les variables d'optimisation du problème NLP sont alors choisies comme étant les valeurs de l'état et des commandes aux points de segmentation  $t_i$ ,  $i = 0, \dots, M$ , i.e.,

$$x = \{x_0, u_0, x_1, u_1, \dots, x_M, u_M\}. \quad (1.111)$$

L'élément fondamental associé aux méthodes de collocation directes consiste à *remplacer* la dynamique globale du système  $\dot{x} = f(x(t), u(t), p, t)$  par un ensemble de *contraintes de résidus*  $\zeta_i = 0$ , devant être satisfaites en chaque point de collocation. Le problème NLP résultant comporte donc l'ensemble de contraintes

$$c(x) = \begin{pmatrix} \psi_0 \\ g(x_0, u_0, p, t_0) \\ \zeta_0 \\ g(x_1, u_1, p, t_1) \\ \zeta_1 \\ \vdots \\ g(x_{M-1}, u_{M-1}, p, t_{M-1}) \\ \zeta_{M-1} \\ g(x_M, u_M, p, t_M) \\ \psi_f \end{pmatrix}, \quad (1.112)$$

où  $\psi_0$  et  $\psi_f$  représentent respectivement les contraintes de bord initiales (1.34) et finales (1.35),  $g(x, u, p, t)$  les contraintes inégalitaires de trajectoire (1.36), et  $\zeta_i$ ,  $i = 1, \dots, M -$

1 les contraintes de résidus entre les segments de trajectoire, définies par exemple par (1.110). Le problème de commande optimale initial (cf définition 1.1) est donc remplacé par un problème de programmation non linéaire de type NLP. De plus, les contraintes (1.34)-(1.36) du problème de commande optimale initial sont remplacées par l'ensemble de contraintes (1.112), vérifiant

$$c_L \leq c(x) \leq c_U.$$

Comme nous l'avons indiqué pour les méthodes de transcription indirectes, les résidus  $\zeta_i$  peuvent être évalués à l'aide des différentes techniques d'intégration implicites de Runge-Kutta décrites dans la section 1.4. Dans un souci d'exhaustivité, nous donnons ci-dessous, dans le cadre de la méthode de collocation directe, l'expression des variables d'optimisation et des résidus pour chacun de ces schémas d'intégration numérique.

### Méthode d'Euler

*Variables d'optimisation :*

$$x = (x_1, u_1, \dots, x_M, u_M).$$

*Résidus :*

$$\zeta_k = x_{k+1} - x_k - h_k f_k. \quad (1.113)$$

### Méthode classique de Runge-Kutta (RK4)

*Variables d'optimisation :*

$$x = (x_1, u_1, \bar{u}_2, \dots, \bar{u}_M, x_M, u_M).$$

*Résidus :*

$$\zeta_k = x_{k+1} - x_k - \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4), \quad (1.114)$$

avec

$$\begin{aligned} k_1 &= h_k f_k, \\ k_2 &= h_k f \left( x_k + \frac{1}{2}k_1, \bar{u}_{k+1}, t_k + \frac{h_k}{2} \right), \\ k_3 &= h_k f \left( x_k + \frac{1}{2}k_2, \bar{u}_{k+1}, t_k + \frac{h_k}{2} \right), \\ k_4 &= h_k f(x_k + k_3, u_{k+1}, t_{k+1}). \end{aligned}$$

### Méthode trapézoïdale

*Variables d'optimisation :*

$$x = (x_1, u_1, \dots, x_M, u_M).$$

*Résidus :*

$$\zeta_k = x_{k+1} - x_k - \frac{h_k}{2}(f_k + f_{k+1}), \quad (1.115)$$



## Méthode de Hermite-Simpson

Variables d'optimisation :

$$x = (x_1, u_1, \bar{u}_2, \dots, \bar{u}_M, x_M, u_M).$$

Résidus :

$$\zeta_k = x_{k+1} - x_k - \frac{h_k}{6}(f_k + 4\bar{f}_{k+1} + f_{k+1}), \quad (1.116)$$

avec

$$\begin{aligned} \bar{x}_{k+1} &= \frac{1}{2}(x_k + x_{k+1}) + \frac{h_k}{8}(f_k - f_{k+1}), \\ \bar{f}_{k+1} &= f\left(\bar{x}_{k+1}, \bar{u}_{k+1}, t_k + \frac{h_k}{2}\right). \end{aligned}$$

Clairement, le problème de collocation directe comprend un nombre important de variables d'optimisation, ce nombre étant approximativement égal à  $n \approx (n_x + n_u)MN$ , où  $n_x$  est la dimension de l'état,  $n_u$  la dimension du vecteur de commandes,  $M$  le nombre de segments de la discrétisation temporelle, et  $N$  le nombre de phases du problème d'optimisation. Heureusement, la propriété de sparsité des matrices jacobienne  $G$  et hessienne  $H$  est largement exploitée par certains solveurs NLP afin de réduire de manière importante la charge de calcul et le stockage des données.

Bien que la méthode de transcription directe permet de contourner la plupart des problèmes rencontrés dans le cadre de l'optimisation de trajectoires, elle présente tout de même certaines limitations. Ces dernières sont inhérentes aux algorithmes d'optimisation NLP de type "creux", notamment en ce qui concerne le calcul des matrices jacobienne et hessienne. Pour les problèmes d'optimisation non convexes, notons que les algorithmes NLP fournissent dans certains cas une solution sous-optimale, c'est à dire représentant seulement un minimum local du problème posé. Finalement, notons que la convergence des solveurs NLP reste relativement sensible au choix de l'estimé de la trajectoire initiale, à partir duquel une solution optimale est recherchée.

La méthode de collocation directe est résumée par l'algorithme 9.

## Applications à l'optimisation de trajectoires de rentrée atmosphérique

L'approche par collocation directe pour l'optimisation de trajectoires de rentrée atmosphérique a donné lieu à un grand nombre de travaux dans le domaine [62, 53, 63, 33] (voir le tableau 1.3 pour trouver d'autres références). Reprenons l'exemple d'optimisation de trajectoires TAEM [60], dont une première approche par tirs multiples a été discutée dans la section 1.5.3. Une approche par inversion dynamique est utilisée ici afin d'améliorer la propriété de convergence du problème d'optimisation. Les équations dynamiques du véhicules sont réécrites en fonction de la variable indépendante  $s$ , qui représente l'abscisse curviligne le long de trace au sol. Le nouveau modèle est alors obtenu en divisant les

**Algorithme 9** : Méthode de transcription directe.

**Entrées** : Variables de décision  $x$ .

**Sorties** : Contraintes  $c(x)$  définies par (1.112).

**Algorithme** :

```

1 tant que les contraintes (1.112) ne sont pas satisfaites faire
2   pour  $k = 1, \dots, N$  faire
3     Initialiser la phase  $k$  : calculer  $\psi_0$  avec (1.34).
4   fin
5   pour  $j = 0, \dots, M - 1$  faire
6     Évaluer la contrainte de trajectoire  $g(x_j^{(k)}, u_j^{(k)}, p^{(k)}, t_j^{(k)})$  avec (1.36), et la
       contrainte de résidu  $\zeta_j^{(k)}$  en utilisant par exemple (1.113), (1.114), (1.115) ou
       (1.116).
7     Calculer la contrainte de trajectoire  $g(x_M^{(k)}, u_M^{(k)}, p^{(k)}, t_M^{(k)})$  avec (1.36), et la
       contrainte de bord  $\psi_f$  avec (1.35).
8   fin
9   Ajuster les variables d'optimisation  $x^{(k)} = \{x_0^{(k)}, u_0^{(k)}, \dots, x_M^{(k)}, u_M^{(k)}\}$ ,
        $k = 1, \dots, N$ , à l'aide d'une méthode d'optimisation de type NLP, de manière
       que les contraintes (1.112) soient satisfaites.
10 fin
11 retourner  $(c(x), F(x))$ .

```

équations (1.95)-(1.100) par  $ds/dt = V \cos \gamma$ , tel que

$$\frac{dx}{ds} = \cos \chi, \quad (1.117)$$

$$\frac{dy}{ds} = \sin \chi, \quad (1.118)$$

$$\frac{dh}{ds} = \tan \gamma, \quad (1.119)$$

$$\frac{dV}{ds} = -\frac{D}{mV \cos \gamma} - \frac{g \tan \gamma}{V}, \quad (1.120)$$

$$\frac{d\gamma}{ds} = \frac{L \cos \mu}{mV^2 \cos \gamma} - \frac{g}{V^2}, \quad (1.121)$$

$$\frac{d\chi}{ds} = \frac{L \sin \mu}{mV^2 \cos^2 \gamma}. \quad (1.122)$$

Les profils de pente  $\gamma(s)$  et de cap  $\chi(s)$  sont paramétrés par des segments de droite régulièrement espacés en fonction de la longueur de la trace au sol  $s$ . Ce paramétrage permet de satisfaire les contraintes aux bords pour les deux angles en fixant la position des noeuds extrêmes aux valeurs désirées. Ensuite, si les dérivées sont approximées par

de simples différences finies

$$\frac{d\gamma}{ds} \approx \frac{\gamma_{i+1} - \gamma_i}{s_{i+1} - s_i}, \quad (1.123)$$

$$\frac{d\chi}{ds} \approx \frac{\chi_{i+1} - \chi_i}{s_{i+1} - s_i}, \quad (1.124)$$

alors les composantes horizontale et verticale de portance ( $L \cos \mu$  et  $L \sin \mu$ ), nécessaires au suivi de la pente de la trajectoire, peuvent être déterminées à partir des équations (1.121) et (1.122). Une fois que la force de portance est connue, la traînée correspondante peut être directement calculée au moyen de la polaire (1.101), et l'équation de vitesse (1.120) peut être intégrée numériquement afin d'obtenir la longueur de la trace au sol à l'instant final  $s_f$ . Les profils des coordonnées de position  $x$ ,  $y$  et  $h$  sont alors obtenus en intégrant les équations (1.117)-(1.119), qui dépendent uniquement de la pente  $\gamma$  et du cap  $\chi$ . La séquence de points de collocation associée aux angles  $\gamma$  et  $\chi$ , ainsi que la longueur de la trace au sol à l'instant final  $s_f$ , sont les variables d'optimisation du problème, choisies de manière à minimiser l'indice de performance

$$J = \sum_{i=1}^{N-1} [(\gamma_{i+1} - \gamma_i)^2 + (\chi_{i+1} - \chi_i)^2], \quad (1.125)$$

où  $N$  représente le nombre de points de collocation, sujet aux contraintes dynamiques de trajectoire (1.117)-(1.120), aux contraintes terminales

$$V(s_f) = V_{ALI}, \quad h(s_f) = h_{ALI}, \quad x(s_f) = x_{ALI}, \quad y(s_f) = 0,$$

et à la contrainte inégalitaire

$$\bar{q}_{LL} \leq \bar{q}(s) \leq \bar{q}_{UL}, \quad (1.126)$$

permettant de contraindre la pression dynamique le long de la trajectoire TAEM.

La fonction de coût (1.125) signifie que l'on cherche à minimiser les taux angulaires dans le plan horizontal et dans le plan vertical, car les points de collocation sur les profils de pente et de cap sont régulièrement espacés le long de la trace au sol. Par conséquent, ces taux angulaires minimaux correspondent essentiellement à des commandes minimales dans le plan horizontal et dans le plan vertical, i.e., à des profils minimaux de gîte et de portance.

Une approche à deux niveaux est utilisée afin de résoudre ce problème d'optimisation et d'en améliorer la convergence. Un premier sous-problème, correspondant à la minimisation de la fonction de coût  $J$ , est d'abord résolu en ignorant les contraintes de vitesse finale (1.120) et de pression dynamique (1.126). Ce problème correspond donc simplement à la satisfaction des contraintes de position (1.117)-(1.119) pour la valeur de la trace au sol à l'instant final  $s_f$ , et ne requiert ni la connaissance des commandes, ni même d'un modèle du véhicule. L'intégration des équations (1.117)-(1.119) est réalisée grâce à une méthode d'intégration trapézoïdale. Ce sous-problème converge rapidement pour un estimé initial

simple des profils des variables d'optimisation (profils linéaires pour  $\gamma(s)$  et  $\chi(s)$ ), et une solution est facilement obtenue en utilisant un algorithme de programmation quadratique séquentielle (SQP) standard.

Ensuite, la solution du sous-problème simplifié précédent est utilisée comme estimation initiale de la solution du problème d'optimisation complet. Pour cette seconde optimisation, on cherche à vérifier les contraintes de vitesse finale et de pression dynamique aux points de collocation. Les équations (1.117)-(1.119) et (1.120) sont intégrées numériquement en utilisant une méthode d'Euler au deuxième ordre.

La limitation de cette approche pour une utilisation embarquée vient principalement du nombre de variables d'optimisation comprises dans le problème de commande optimale. En première approximation, plus le nombre de variables d'optimisation tend à être grand, plus le temps CPU nécessaire pour résoudre le problème de commande optimale sera important [64]. Dans le chapitre 2, nous verrons que l'utilisation de la propriété de platitude du modèle dans le cadre de l'optimisation de trajectoires de rentrée permet de minimiser le nombre de variables d'optimisation, réduisant alors le temps de calcul associé de manière significative.

### 1.5.6 Quelques autres méthodes

De nombreuses autres méthodes d'optimisation de trajectoires permettent de résoudre les problèmes de commande optimale, comme les méthodes "intelligentes", c'est à dire les réseaux de neurones, les algorithmes évolutionnistes où les moteurs d'inférence. Ces techniques sont souvent jugées comme étant moins compétitives que les méthodes d'optimisation classiques décrites dans les sections 1.5.1 à 1.5.5, car beaucoup d'entre elles n'exploitent pas l'information de gradient. Cependant, les méthodes intelligentes présentent l'avantage de ne nécessiter que d'une connaissance partielle du problème d'optimisation à traiter.

#### Réseaux de neurones

Les réseaux de neurones non bouclés à apprentissage supervisé sont bien connus pour être des approximateurs universels parcimonieux [65]. Ils constituent un outil efficace pour la modélisation, l'estimation et la commande de processus. Le lecteur trouvera plus de détails sur les techniques de modélisation et de commande par réseaux de neurones dans l'ouvrage [66].

La propriété de parcimonie des réseaux de neurones peut être utilisée de manière avantageuse afin de modéliser de manière statique un ensemble de trajectoires admissibles, préalablement planifiées vis-à-vis d'un ensemble de scénarios prédéfinis. Le faisceau de trajectoires est alors modélisé par une application non linéaire continue. Le réseau de neurones associé peut alors être interrogé en ligne, afin d'interpoler dans la base de données ainsi créée, la trajectoire de référence associée à des conditions opérationnelles données.

Cette approche “boîte noire” a notamment été utilisée dans le cadre de l’optimisation de trajectoires de rentrée atmosphérique pour la phase d’atterrissage automatique du véhicule X-34. Dans [67], la méthodologie de planification de trajectoires en ligne proposée comprend principalement trois étapes :

1. La génération “hors ligne” d’une base de trajectoires optimales, valides pour un ensemble de conditions opérationnelles données. Les conditions d’optimalité associées au principe du maximum de Pontryagin sont alors utilisées pour formuler un problème d’optimisation de type indirect,
2. L’apprentissage des trajectoires planifiées au cours de l’étape 1, chacune d’entre elle étant complètement caractérisée par la donnée d’une paire d’états/co-états. Des réseaux de neurones polynomiaux sont alors utilisés pour apprendre les relations existant entre les observables du véhicule (par ex. les états, les capacités opérationnelles) et les co-états définis en différents points de l’enveloppe de vol,
3. La dernière étape est effectuée à bord du véhicule : les trajectoires sont replanifiées en vol en fonction de la dynamique du véhicule. L’état courant du véhicule est alors estimé afin de déterminer les co-états associés, par interpolation dans le réseau de neurones construit à l’étape 2. Avec ces valeurs d’états et de co-états, la trajectoire de vol courante est alors générée en propageant la dynamique du véhicule et les équations adjointes sur un intervalle de temps prédéfini.

### Algorithmes évolutionnistes

Les méthodes évolutionnistes, telles que les algorithmes génétiques, les techniques de recuit simulé, d’essais particuliers ou encore les méthodes de type Monte Carlo, sont des méthodes de recherche *globales*, qui consistent à faire évoluer les valeurs des variables d’optimisation (inconnues) de manière probabiliste<sup>28</sup>. Le lecteur intéressé est renvoyé à l’ouvrage [68] pour obtenir davantage de détails sur le sujet. Un indice de coût est alors associé à chaque solution (trajectoire) candidate. Après un nombre donné d’itérations probabilistes sur les variables d’optimisation, la solution candidate possédant le meilleur coût est alors choisie comme solution optimale. Le nombre de calculs nécessaires à l’obtention d’une trajectoire optimale<sup>29</sup> est souvent prohibitif pour qu’un algorithme évolutionniste puisse être utilisé à bord d’un véhicule.

Les algorithmes génétiques ont été proposés comme solution globale au problème d’optimisation de trajectoires de rentrée atmosphérique dans [69]. L’approche consiste d’abord à discrétiser les commandes du modèle dynamique du véhicule (angles d’incidence et de gîte) sur un intervalle de temps donné, puis à déterminer les valeurs optimales de ces commandes en un nombre fini de points de collocation. Un individu est alors constitué de l’ensemble des variables d’optimisation permettant de décrire les profils des commandes

---

<sup>28</sup>de nombreuses heuristiques peuvent être incluses lors du choix successif des variables, afin de “guider” plus ou moins efficacement la recherche vers la solution optimale.

<sup>29</sup>dans la plupart des cas, les algorithmes évolutionnistes fournissent seulement une solution sous-optimale au problème d’optimisation, car le critère d’arrêt est souvent choisi de manière heuristique, par exemple par rapport au nombre de générations de solutions candidates produites.

et est associé à chaque trajectoire candidate. Le problème de planification de trajectoires est ensuite résolu “hors ligne”, sur un nombre de générations donné, par croisements et mutations successives entre les différents individus (boucle génétique).

Dans ce mémoire, nous utiliserons un algorithme génétique dans le cadre d’une stratégie de convexification du problème de commande optimale par des formes géométriques déformables, appelées superquadriques (voir section 2.4).

### Moteurs d’inférence

Les systèmes de commande floue ont émergé au milieu des années 60, peu après le développement de la théorie des ensembles flous par L. Zadeh [70]. Le lecteur intéressé pourra trouver une introduction très complète aux techniques de commande floue dans [71]. Un système classique de commande floue comporte principalement quatre éléments :

- un *mécanisme de fuzzyfication*, permettant de quantifier les variables linguistiques par des fonctions d’appartenance de type floues,
- une *base de connaissance*, représentée par un ensemble de règles linguistiques du type “Si-Alors”, qui permet de modéliser la connaissance d’un expert humain sur le bon fonctionnement d’un système en boucle fermée,
- un *moteur d’inférence* qui détermine les règles actives de la base connaissance pour un état donné du système,
- un *mécanisme de défuzzyfication*, qui convertit les règles actives en commandes physiques à appliquer au système.

Les techniques de commande floue ont été appliquées au problème de planification en ligne de trajectoires TAEM dans [72]. Connaissant l’état du véhicule, i.e., son niveau d’énergie et l’état de ses surfaces de commande aérodynamiques, des moteurs d’inférence permettent alors de déterminer les positions optimales des points géométriques décrivant la trajectoire TAEM (par exemple, le rayon du virage d’enroulement autour du HAC et la position longitudinale du HAC par rapport au seuil de piste).

## 1.6 Conclusion

Pour conclure ce chapitre, rappelons brièvement les différentes méthodes numériques permettant de résoudre un problème de programmation non linéaire (NLP), ainsi que les techniques d’optimisation de trajectoires permettant de transcrire un problème de commande optimale (OCP) en un problème NLP de dimension finie.

La plupart des approches permettant de résoudre un problème de programmation non linéaire sont basées sur la méthode d’optimisation de Newton. A l’heure actuelle, les méthodes les plus répandues dans la communauté scientifique reposent sur des algorithmes de programmation quadratique séquentielle (SQP), visant à résoudre de manière itérative un sous-problème de programmation quadratique (QP).

Les deux grandes approches utilisées pour transformer un problème d'optimisation de trajectoires en un problème de programmation non linéaires sont :

- Les *méthodes de tirs*, consistant à résoudre un problème aux deux bouts, c'est à dire à déterminer les conditions initiales de la trajectoire, de telle sorte que la propagation numérique de cette dernière satisfasse un ensemble de contraintes de bord à l'instant final,
- Les *méthodes de transcription*, basées sur la conversion du problème de commande optimale en un problème de dimension finie, dans lequel les contraintes sont vérifiées en un certain nombre de points, appelés points de collocation.

Les *méthodes indirectes* reposent sur une application directe du principe du maximum de Pontryagin, alors que les *méthodes directes* utilisent les variables de dimensionnement de la trajectoire comme variables d'optimisation du problème NLP.

Le tableau 1.3 présente un certain nombre d'applications de ces techniques dans le cadre de l'optimisation de trajectoires pour les véhicules de rentrée atmosphérique.

Méthode	Références
Tirs simples directs	[73],[50],[74],[75],[51],[76],[52],[53],[77],[78],[79],[54],[80],[81],[82],[83],[84],[37],[85],[86]
Tirs simples indirects	[36],[38]
Tirs multiples	[56],[60],[58],[87],[88],[59],[89],[57],[90],[91],[92],[93],[94],[95],[96],[97]
Transcription indirecte	[61]
Transcription directe	[62],[98],[53],[77],[99],[63],[33],[100],[101],[102],[12],[103],[104],[105],[106],[60]
Réseaux de neurones (indirects)	[72],[107],[108],[35],[109],[110],[111]
Algorithmes génétiques (GA)	[112],[69],[113],[114],[115]
Logique floue (FL)	[116],[67],[117],[118]
Approches hybrides (GA/FL)	[119],[67]
Méthode du point intérieur	[120]

TAB. 1.3 – classification des principales techniques d'optimisation de trajectoires de rentrée atmosphérique.

Comme nous l'avons souligné, les techniques de planification de trajectoires de rentrée atmosphérique par tirs simples ou multiples nécessitent le plus souvent un partitionnement géométrique de la trajectoire de vol en un certain nombre de segments sur lesquels la dynamique du véhicule est approximée. Les différentes phases de vol (par exemple virage à inclinaison constante, trajectoire de descente rectiligne à pente constante, ...) sont ensuite mises bout à bout, et les paramètres géométriques de dimensionnement sont obtenus par itérations successives, en propageant la trajectoire de l'instant initial à l'instant final. L'artifice de segmentation utilisé est à l'origine de l'efficacité numérique de la méthode.

En effet, le nombre réduit de paramètres nécessaires pour décrire complètement la trajectoire de vol engendre un temps de calcul relativement faible. Cependant, ce même artifice est également à l'origine d'erreurs de modèles propagées le long de la trajectoire, car la dynamique du modèle à trois degrés de liberté du véhicule n'est pas vérifiée de manière exacte.

Les techniques de transcription directe et indirecte utilisent une paramétrisation des états (ou des co-états) et des commandes du système par des courbes candidates, par exemple des polynômes par morceaux ou des courbes de Bézier. Dans le cadre de la planification de trajectoires de rentrée atmosphérique, la dynamique du véhicule et les contraintes de trajectoire sont alors vérifiées sur une séquence de points de collocation. Dans ce cas, si la séquence des points de collocation est suffisamment dense, la trajectoire optimale obtenue vérifie exactement la dynamique du système, mais au prix d'un temps de calcul plus élevé que pour les techniques de tirs.

Ce tour d'horizon des différentes techniques de génération de trajectoires permet de positionner nos travaux et de justifier les orientations suivies par la suite. Dans le chapitre suivant, nous développerons pas à pas une loi de guidage autonome, incluant la replanification de trajectoires à bord, pour les véhicules de rentrée atmosphérique sujets à des défaillances de surfaces de contrôle aérodynamique. L'utilisation de la propriété de platitude du modèle, couplée à une méthode de collocation directe, nous permettra d'obtenir des trajectoires de rentrée vérifiant de manière exacte la dynamique du véhicule. En outre, les trajectoires pourront être générées en quelques dizaines de millisecondes, la dynamique du système n'apparaissant plus de manière explicite dans le problème de commande optimale.





# Chapitre 2

## Méthodologie proposée

“Il est doux à tout âge de se laisser guider par la fantaisie.”

Marcel Proust

---

### Résumé :

*Dans ce chapitre, nous proposons une méthodologie originale pour le développement d'une loi de guidage autonome.*

*Dans un premier temps, nous présentons l'architecture de la loi de guidage autonome retenue dans le cadre de cette étude.*

*Ensuite, la propriété de platitude du modèle du véhicule est utilisée afin de générer des trajectoires en un temps réduit.*

*Une méthodologie d'annexion superellipsoïdale est également développée afin de rendre le problème de commande optimale, formulé dans l'espace de la sortie plate, convexe. En outre, nous proposons un certain nombre de résultats nouveaux portant sur ces formes géométriques en dimension  $n$ .*

*Une méthode originale, permettant de replanifier des trajectoires en présence de défauts actionneurs, est ensuite présentée.*

*La dernière section est dédiée à la transformation du problème de commande optimale en un problème de programmation non linéaire.*

---

## 2.1 Introduction

Dans ce chapitre, nous développons étape par étape une loi de guidage autonome pour les véhicules de rentrée atmosphérique. Cette loi de guidage comprend un planificateur de trajectoires en ligne tolérant aux défauts (FTOPP), un bloc d'inversion dynamique (NDI) ainsi qu'un correcteur de suivi de trajectoires de type linéaire à paramètres variants (LPV). La stratégie proposée tient compte de l'effet que peuvent avoir les défaillances des surfaces de contrôle aérodynamique sur le modèle du véhicule à trois degrés de liberté. Nous supposons que des sorties fiables d'un module de détection et de localisation de défaillances (FDI) sont disponibles au planificateur de trajectoires. Sous cette hypothèse, une nouvelle trajectoire admissible, appartenant à l'enveloppe de vol réduite du véhicule, est alors calculée en ligne afin de tenir compte d'une situation défaillante, sans avoir recours à une banque de trajectoires avoisinantes pré-déterminées. Un bénéfice direct est que le segment sol est exempté d'un grand nombre de tâches opérationnelles "bas niveau", pouvant alors mieux assurer son rôle de coordination globale.

**Remarque 2.1.** Sous sa forme actuelle, l'algorithme de replanification de trajectoires proposé est en mesure de traiter des situations défaillantes de type blocages, départs en butée, et perte d'efficacité de gouvernes. De plus, d'après les phases de vol considérées (dissipation d'énergie en zone terminale, et atterrissage automatique), nous ne considérons pas dans cette étude les défaillances des tuyères du système de contrôle d'attitude (RCS). Nous ne traitons pas non plus les pannes de type oscillatoires.

L'algorithme de replanification de trajectoires est basé sur les techniques de transcription directe et la résolution d'un problème de programmation non linéaire (NLP), tenant compte de la propriété de platitude du modèle à 3 ddl du véhicule. En outre, des contraintes de trajectoire additionnelles, résultant de défaillances simples ou multiples de gouvernes, sont incluses dans le problème de commande optimale (OCP) afin que le véhicule dégradé puisse être équilibré<sup>30</sup> statiquement le long de la trajectoire replanifiée.

L'avantage principal de cette approche par rapport aux techniques existantes est que le problème de commande optimal initial est transcrit dans un cadre géométrique par l'utilisation de la propriété de platitude du modèle. Ceci conduit à un nouveau problème de commande optimale dépourvu de contraintes dynamiques et pouvant être résolu plus rapidement que dans l'espace d'état. Dans ce cadre, les coefficients aérodynamiques sont représentés sous la forme de fonctions analytiques différentiables, gouvernées par un petit nombre de paramètres. De plus, afin d'améliorer la convergence du processus d'optimisation, les contraintes de trajectoire associées au problème de guidage sont convexifiées par des formes géométriques déformables, appelées *superquadriques*. Dans le chapitre 3, nous verrons également qu'il est possible de tenir compte de certaines perturbations aérologiques dans l'algorithme de planification de trajectoires à bord. La méthodologie proposée a été appliquée avec succès à un nombre prédéfini de scénarii défaillants pouvant survenir lors des phases de vol critiques d'un véhicule de rentrée atmosphérique, en utilisant un

---

<sup>30</sup>on parle d'*équilibre statique* lorsque le pilote n'a pas d'action à effectuer tant que l'engin est sur la trajectoire. Nous utiliserons indifféremment les termes d'"équilibre" et de "trimmabilité" dans la suite du manuscrit, le dernier terme étant largement utilisé dans le domaine aéronautique et spatial.

simulateur<sup>31</sup> haute fidélité, développé dans le cadre de cette étude pour la navette spatiale américaine STS-1.

Les trajectoires étant générées à bord du véhicule, il est primordial que le régulateur de poursuite de trajectoires puisse également s'adapter afin que le système conserve un certain niveau de robustesse et de performances. Nous avons choisi pour cela un régulateur linéaire à paramètres variants, piloté par la trajectoire de vol courante. Le régulateur est synthétisé dans le formalisme des transformations linéaires fractionnaires (LFT).

Nous nous sommes efforcés de décrire la démarche proposée dans un contexte suffisamment général pour qu'elle soit facilement transposable à d'autres types de missions et/ou d'autres types de véhicules (par exemple, missions de surveillance par des drones, manœuvres d'attitude de satellites sur orbite, trajectoires guidées de missiles, ...). L'application de cette démarche à une mission de rentrée atmosphérique sera traitée en détails dans le chapitre 3.

## 2.2 Architecture générale de la loi de guidage autonome

L'architecture générale de la loi de guidage autonome proposée est illustrée sur la figure 2.1. Ce diagramme fonctionnel peut être séparé en deux parties distinctes : une couche de supervision, permettant de détecter, localiser et d'identifier des défaillances, et une chaîne de guidage, pilotage et navigation (GNC).

Le premier sous-ensemble représente une architecture GNC reconfigurable incluant des fonctions de guidage, pilotage et de navigation. Le module de guidage autonome comprend un *planificateur de trajectoires* en ligne tolérant aux défauts (FTOPP), permettant de calculer des trajectoires de rentrée admissibles à bord du véhicule en présence d'une situation défaillante, un *bloc d'inversion dynamique* (NDI) fournissant les consignes de guidage nominales en fonction de la sortie du planificateur de trajectoires, et un *régulateur de poursuite* de trajectoires, permettant d'offrir un niveau de robustesse et de performances donné à la chaîne GNC.

Le planificateur de trajectoires est alimenté par les données de mission (voir figure 2.1), telles que la localisation du point de transition avec la phase d'atterrissage automatique (point ALI) et la cinématique désirée en ce point, mais également les sites d'atterrissage de secours en cas d'anomalie à bord et abandon de mission (cas des phases "abort"). De plus, les contraintes de mission constituent également une entrée du planificateur. Par exemple, les contraintes mécaniques (facteur de charge, pression dynamique et flux thermique pour la phase hypersonique) liés à la conception du véhicule, ou encore les contraintes de trajectoire associées à l'interdiction de survol de certaines régions habitées, entrent dans cette

---

<sup>31</sup>ce simulateur, développé par l'auteur, est disponible sur simple demande à l'adresse [vincent.morio@ims-bordeaux.fr](mailto:vincent.morio@ims-bordeaux.fr), [franck.cazaurang@ims-bordeaux.fr](mailto:franck.cazaurang@ims-bordeaux.fr) ou [ali.zolghadri@ims-bordeaux.fr](mailto:ali.zolghadri@ims-bordeaux.fr).

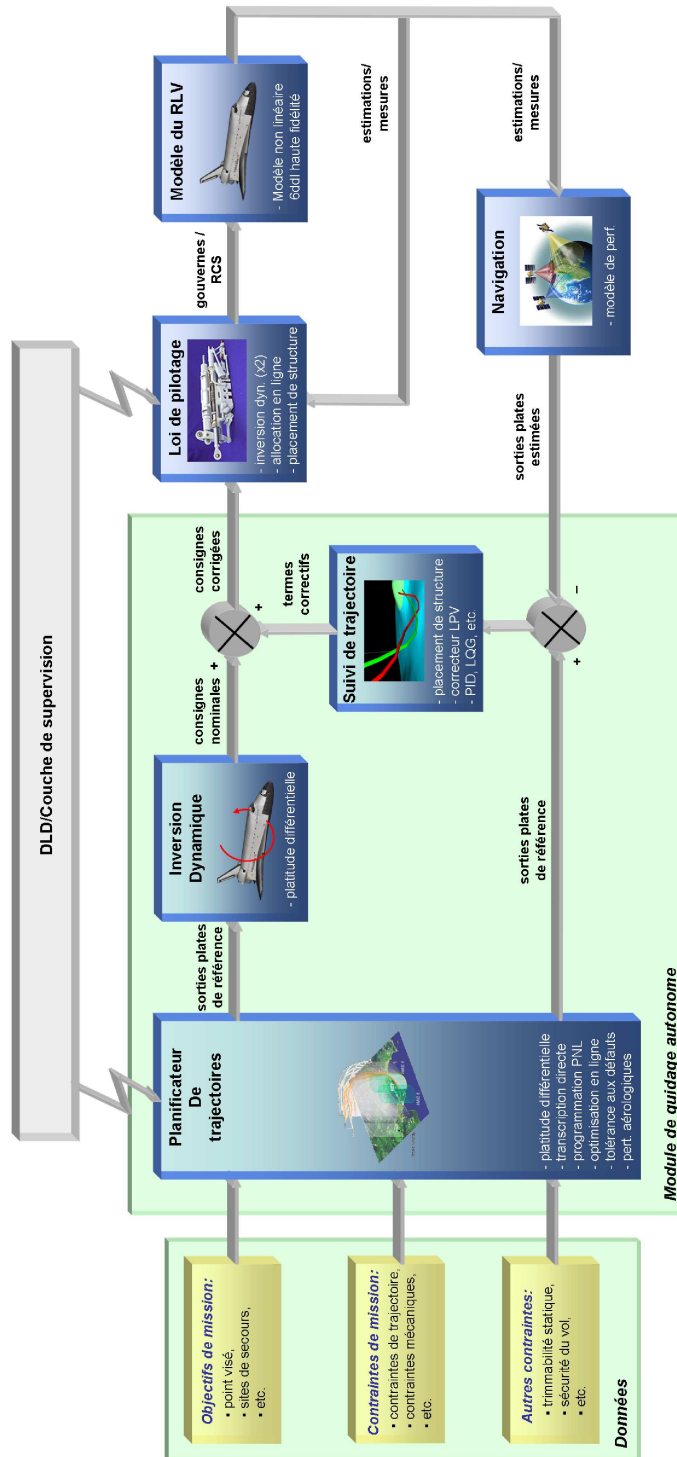


FIG. 2.1 – architecture de la loi de guidage autonome.

catégorie. La dernière entrée du planificateur comprend les contraintes auxiliaires, telles que les contraintes d'équilibrage statique du véhicule en situation défaillante, ainsi que les contraintes liées à la sécurité du vol (taux de virage maximum, marges de décrochage, limitations, etc.).

Du fait de la prise en compte de la platitude du modèle lors de la planification de trajectoires, le bloc d'inversion dynamique non linéaire (NDI) du module de guidage autonome, représenté sur la figure 2.1, comprend uniquement des expressions analytiques afin de déterminer les consignes de guidage nominales correspondant aux profils de référence des sorties plates générées par le planificateur.

Enfin, le régulateur de suivi de trajectoires doit être capable de s'adapter à toute trajectoire admissible générée à bord du véhicule, afin de garantir un niveau de robustesse et de performances donné. Un choix naturel consiste alors à synthétiser ce régulateur sous la forme d'un correcteur linéaire à paramètres variants, les paramètres variant étant alors choisis comme étant représentatifs de la trajectoire de rentrée courante. La synthèse est réalisée dans le formalisme des transformations linéaires fractionnaires (LFT), en utilisant la démarche proposée dans [121]. Des premiers résultats ont été obtenus dans [122] pour la phase de rentrée hypersonique de la capsule européenne ARD (Atmospheric Reentry Demonstrator). La synthèse du correcteur reposant sur des outils LPV classiques [121, 123], nous avons choisi de ne pas la détailler dans ce mémoire et de focaliser notre attention sur les sous-fonctions de planification de trajectoires à bord et d'inversion dynamique. Finalement, la synthèse d'un régulateur non linéaire de poursuite de trajectoires, basé sur la théorie des systèmes plats, a notamment été proposée dans [124, 125, 126, 127] pour la phase de rentrée hypersonique du véhicule européen IXV (Intermediate eXperimental Vehicle) ainsi que pour la phase de dissipation d'énergie en zone terminale du véhicule planant européen ARES (Atmospheric Reentry Experimental Spaceplane).

L'objectif de la loi de pilotage est de déterminer les couples à appliquer au véhicule, en fonction des consignes corrigées provenant du module de guidage autonome, et à les allouer aux ressources de commande disponibles (gouvernes aérodynamiques et/ou système RCS). Une architecture non linéaire de pilotage à "deux échelles de temps" a été développée dans le cadre de ces travaux (voir figure 2.2) pour la navette spatiale américaine STS-1. Les couples nominaux sont alors déterminés par inversion dynamique non linéaire des modèles de boucle interne (vitesses  $p$ ,  $q$  et  $r$ ) et de boucle externe (couples  $L$ ,  $M$  et  $N$ ), résultant de l'hypothèse de découplage des dynamiques lente<sup>32</sup> et rapide [128, 129]. Les correcteurs associés à la boucle interne et à la boucle externe sont synthétisés par placement de structure propre à partir des modèles aux petites variations, de type linéaires à temps variant (LTV), associés à chaque triplet d'équations dynamiques de pilotage [130]. Finalement, l'allocation de gouvernes<sup>33</sup> est réalisée en ligne par la résolution d'un pro-

---

<sup>32</sup>la dynamique associée aux angles  $\alpha$ ,  $\beta$  et  $\mu$  est qualifiée de dynamique "lente" car l'effet des commandes (gouvernes et/ou système RCS) sur ces équations est plus faible que sur les équations régissant l'évolution des vitesses  $p$ ,  $q$  et  $r$ .

<sup>33</sup>du fait des trajectoires de rentrée considérées dans cette étude (TAEM et A&L), nous n'utiliserons pas les degrés de liberté supplémentaires associés au système RCS.

blème de programmation non linéaire, les coefficients de moments aérodynamiques étant approximés sous la forme de fonctions non linéaires différentiables grâce à des réseaux de neurones analytiques (perceptrons multi-couches). Les braquages de gouvernes calculés par le module d'allocation sont alors appliqués à un modèle à six degrés de liberté "haute fidélité" du véhicule de rentrée atmosphérique considéré (voir figure 2.1).

Finalement, le module de navigation (cf figure 2.1) permet d'estimer la position et l'attitude du véhicule à chaque instant. Dans le cadre de cette étude, le module de navigation contient uniquement des modèles de performance associés au système inertiel/GPS, l'objectif étant bien sûr d'analyser les performances et la robustesse de la loi de guidage autonome vis-à-vis de situations défailtantes potentielles.

L'ensemble de la chaîne GNC est supervisée par un module de détection, localisation et reconfiguration (FDIR), qui agit sur la chaîne GNC durant la mission de rentrée afin d'assigner un niveau d'autonomie opérationnelle et/ou décisionnelle au véhicule. Un tel niveau d'autonomie peut être atteint en effectuant une surveillance continue de l'état de santé du véhicule (par exemple en utilisant une fonction FDI), puis par une reconfiguration d'une partie ou de l'ensemble de la chaîne GNC. Dans le cadre de notre étude, nous traitons uniquement les aspects replanification de trajectoires à bord et adaptation de la boucle de guidage, la loi de pilotage comportant déjà un module d'allocation tolérant aux défauts (cf figure 2.2).

**Remarque 2.2.** En plus de la fonction de détection et de localisation de défaillances, nécessaire pour indiquer qu'un événement imprévu s'est produit ainsi que les canaux d'action impliqués, la capacité d'identifier/quantifier les défauts représente une démarche importante dans le cadre de la replanification de trajectoires tolérante. La fonction d'identification de défauts permet alors d'indiquer, par exemple, la position d'une gouverne bloquée qui constituera une entrée du planificateur de trajectoires (voir figure 2.1).

L'ensemble du processus associé au développement du planificateur de trajectoires tolérant aux défauts est représenté sur l'organigramme 2.3. Conceptuellement, la méthodologie de planification de trajectoires proposée comprend les étapes suivantes :

1. La dynamique non linéaire du système est transcrite dans un espace de dimension plus faible afin de réduire le nombre de variables d'optimisation. Ceci est réalisé en s'appuyant sur la propriété de platitude différentielle du système.
2. Les contraintes d'équilibrage statique associées au modèle à six degrés de liberté du véhicule sont ensuite introduites dans les équations à trois degrés de liberté de façon à ce que les trajectoires générées lors de l'apparition d'une ou plusieurs pannes actionneurs restent à l'intérieur de l'enveloppe de vol réduite.
3. Une fois transcrites dans l'espace des sorties plates, les contraintes de trajectoire présentes dans le problème de commande optimale (OCP) peuvent devenir non convexes [64]. Nous allons alors chercher à approximer ces contraintes par des formes convexes déformables lisses, en utilisant un outil logiciel développé à cette fin par l'auteur sous l'environnement Matlab, appelé OCEANS (Optimal Convexification by Evolutionary Algorithms aNd Superquadrics). Nous obtiendrons alors un problème

2.2. Architecture générale de la loi de guidage autonome

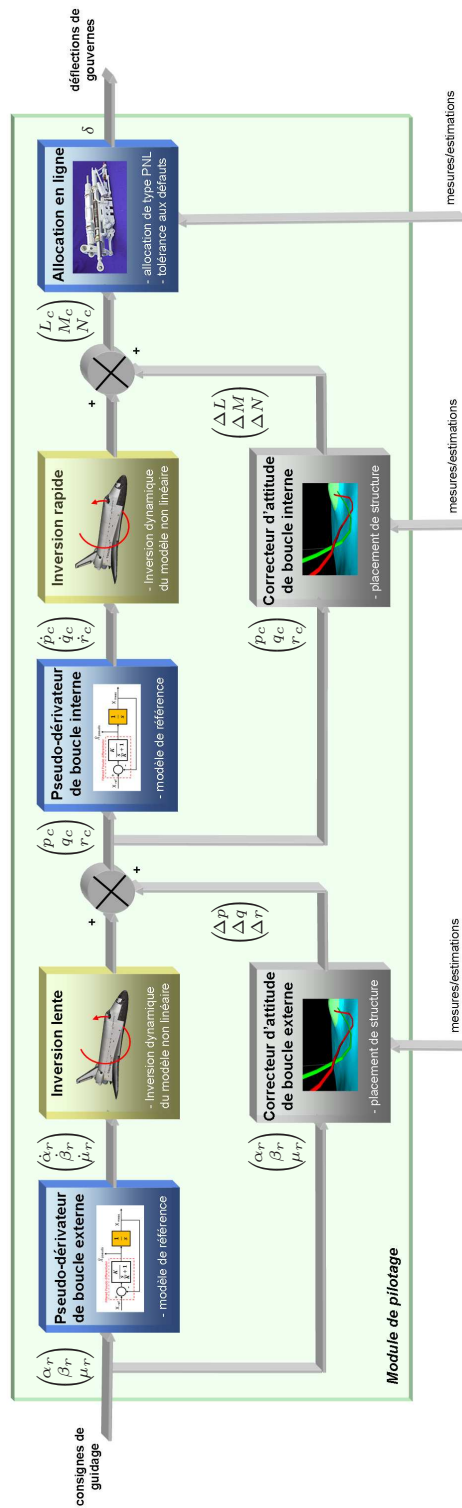


FIG. 2.2 – architecture de la loi de pilotage non linéaire.



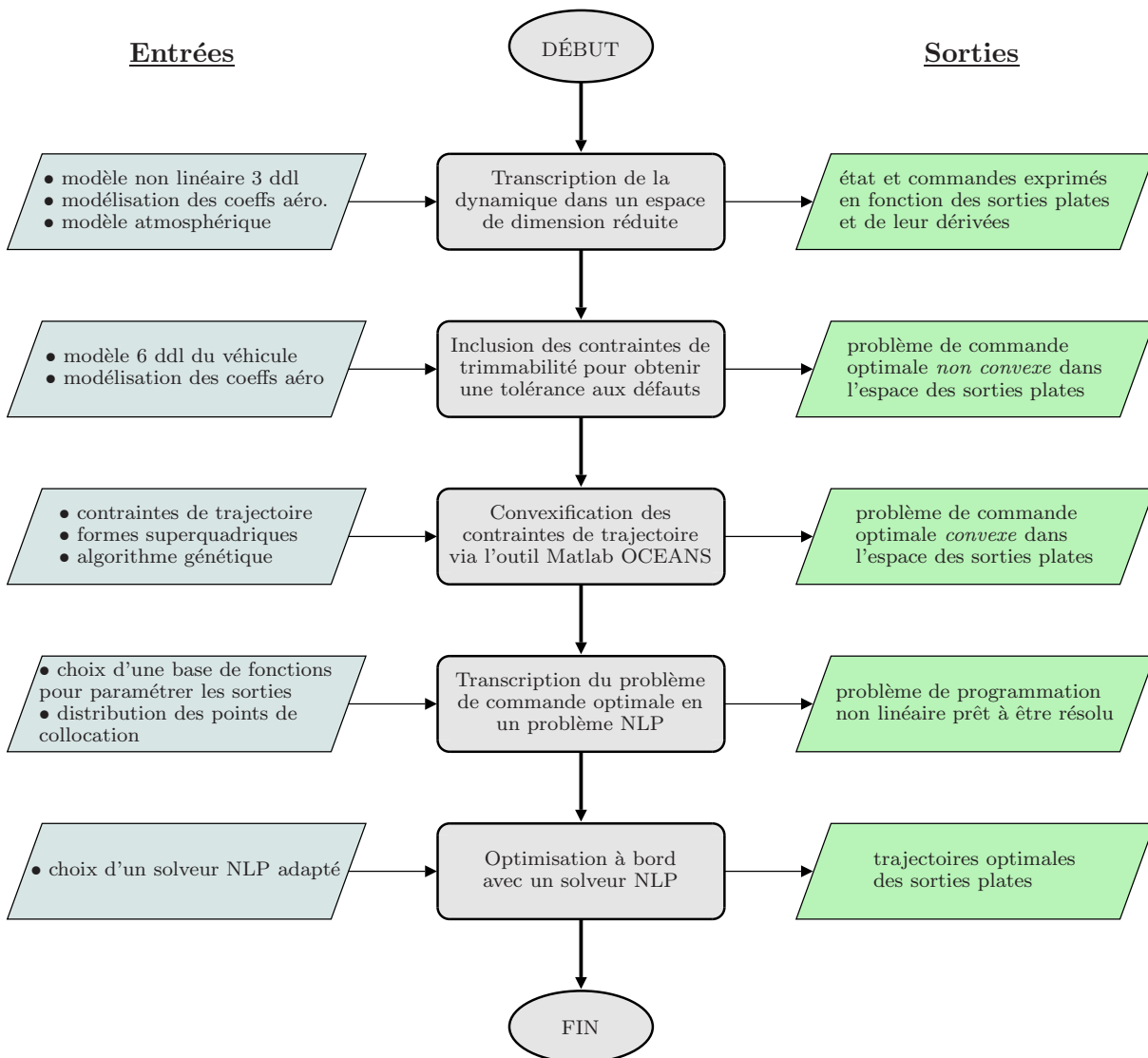


FIG. 2.3 – organigramme illustrant la méthodologie de planification de trajectoires proposée.

de commande optimale qui sera plus conservateur, mais qui aura la propriété forte d'être convexe. Ainsi, nous pourrons garantir la convergence du solveur lorsque la trajectoire de vol recherchée se situe à l'intérieur des formes superquadriques convexes.

4. L'OCP est ensuite transformé en un problème de programmation non linéaire (NLP) en utilisant une technique de transcription directe, après que les variables d'optimisation du problème aient été paramétrées par des B-splines, formes particulières de polynômes par morceaux.
5. Le problème NLP est finalement résolu à bord du véhicule en utilisant un solveur de programmation non linéaire adapté.

**Remarque 2.3.** Une fois que les profils optimaux des sorties plates sont connus, la génération des consignes de guidage le long de la trajectoire est réalisée par le bloc d’inversion dynamique non linéaire (NDI) représenté sur la figure 2.1. Ce bloc contient principalement les isomorphismes reliant l’état et les commandes du système aux sorties plates et à leurs dérivées successives.

Dans les sections suivantes, nous allons détailler pas à pas chacune des étapes associées au développement du planificateur de trajectoires tolérant aux défauts. Ainsi, nous commencerons par rappeler comment la propriété de platitude d’un système peut être utilisée dans le cadre de la planification de trajectoires à bord, afin de réduire le nombre de variables d’optimisation intervenant dans le problème considéré. Ensuite, nous présenterons une méthode d’annexion superellipsoïdale, permettant de convexifier le problème de commande optimale obtenu à l’étape précédente. La prise en compte des contraintes d’équilibrage statique, permettant de rendre le planificateur tolérant aux défauts, sera étudiée dans la section 2.5. Finalement, nous verrons comment transformer le problème de commande optimale, décrit dans l’espace de la sortie plate, en un problème de programmation non linéaire, en utilisant une méthode de transcription directe.

## 2.3 Planification de trajectoires par platitude

Dans cette section, nous abordons le problème de planification de trajectoires par platitude. Nous commencerons par décrire de manière générale les problèmes de commande optimale considérés dans le cadre de cette étude. Ensuite, nous rappellerons quelques définitions usuelles concernant les systèmes plats, afin d’améliorer la lisibilité de la démarche. Finalement, nous utiliserons la propriété de platitude du modèle afin de transcrire le problème de commande optimale initial dans l’espace de la sortie plate.

### 2.3.1 Classe de problèmes de commande optimale considérée

Considérons un système non linéaire déterministe défini sur une variété différentiable  $X$ , et s’écrivant sous la forme

$$\begin{cases} \dot{x}(t) = f(x(t), u(t)), \\ y(t) = h(x(t), u(t)), \end{cases} \quad (2.1)$$

où  $t$  représente la variable temporelle,  $x : [t_0, t_f] \mapsto \mathbb{R}^n$  est le vecteur d’état de dimension  $n$ ,  $u : [t_0, t_f] \mapsto \mathbb{R}^m$  est le vecteur de commandes de dimension  $m$ , et  $y : [t_0, t_f] \mapsto \mathbb{R}^p$  représente le vecteur des sorties mesurées/estimées de dimension  $p$ . De plus,  $f(\cdot)$  et  $h(\cdot)$  représentent des applications non linéaires suffisamment lisses sur  $X$ .

Nous supposons que l’ensemble des objectifs associés au planificateur de trajectoires, définis soit au niveau “mission” ou au niveau “véhicule”, s’intègre de manière naturelle dans un problème de commande optimale (OCP) défini dans l’espace d’état. Nous cherchons ainsi à déterminer une trajectoire du système (2.1) dans  $[t_0, t_f]$ , permettant de minimiser

la fonctionnelle de coût<sup>34</sup>  $J : \mathbb{R}^m \times \mathbb{R}^+ \rightarrow \mathbb{R}$ , telle que

$$J \triangleq \mathcal{C}_0(x(t_0), u(t_0), t_0) + \int_{t_0}^{t_f} \mathcal{C}_t(x(t), u(t), t) dt + \mathcal{C}_f(x(t_f), u(t_f), t_f), \quad (2.2)$$

sujette à un ensemble de contraintes linéaires ( $N_0$  contraintes initiales,  $N_f$  contraintes finales et  $N_t$  contraintes de trajectoire) et non linéaires ( $N'_0$  contraintes initiales,  $N'_f$  contraintes finales et  $N'_t$  contraintes de trajectoire), telles que

$$\begin{aligned} l_0 &\leq A_0 x(t_0) + B_0 u(t_0) \leq u_0, & A_0 &\in \mathbb{R}^{N_0 \times n}, & B_0 &\in \mathbb{R}^{N_0 \times m}, \\ l_t &\leq A_t x(t) + B_t u(t) \leq u_t, & A_t &\in \mathbb{R}^{N_t \times n}, & B_t &\in \mathbb{R}^{N_t \times m}, & t \in [t_0, t_f], \\ l_f &\leq A_f x(t_f) + B_f u(t_f) \leq u_f, & A_f &\in \mathbb{R}^{N_f \times n}, & B_f &\in \mathbb{R}^{N_f \times m}, \\ L_0 &\leq c_0(x(t_0), u(t_0)) \leq U_0, \\ L_t &\leq c_t(x(t), u(t)) \leq U_t, & t &\in [t_0, t_f], \\ L_f &\leq c_f(x(t_f), u(t_f)) \leq U_f. \end{aligned} \quad (2.3)$$

On suppose en outre que les fonctions  $c_0 : \mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R}^{N'_0}$ ,  $c_f : \mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R}^{N'_f}$  et  $c_t : \mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R}^{N'_t}$  sont au moins de classe  $C^2$  sur des ouverts denses convenablement choisis. De plus, la durée totale de la trajectoire  $t_f$  peut soit être fixée, soit être libre.

En résumé, nous cherchons à résoudre le problème de commande optimale suivant :

$$\begin{aligned} \min_{x(t), u(t)} & \mathcal{C}_0(x(t_0), u(t_0), t_0) + \int_{t_0}^{t_f} \mathcal{C}_t(x(t), u(t), t) dt + \mathcal{C}_f(x(t_f), u(t_f), t_f) \quad (2.4) \\ \text{t.q.} & \end{aligned}$$

$$\begin{aligned} \dot{x}(t) &= f(x(t), u(t)), & t &\in [t_0, t_f], \\ l_0 &\leq A_0 x(t_0) + B_0 u(t_0) \leq u_0, \\ l_t &\leq A_t x(t) + B_t u(t) \leq u_t, & t &\in [t_0, t_f], \\ l_f &\leq A_f x(t_f) + B_f u(t_f) \leq u_f, \\ L_0 &\leq c_0(x(t_0), u(t_0)) \leq U_0, \\ L_t &\leq c_t(x(t), u(t)) \leq U_t, & t &\in [t_0, t_f], \\ L_f &\leq c_f(x(t_f), u(t_f)) \leq U_f. \end{aligned}$$

### 2.3.2 Rappels sur le concept de platitude différentielle

Avant d'utiliser le concept de platitude dans le cadre de la planification de trajectoires, nous rappelons ici quelques définitions de base concernant les systèmes plats. Ces définitions sont nécessaires pour camper le décor et permettre au lecteur de suivre les développements méthodologiques de ce chapitre, sans avoir systématiquement recours à la première partie du manuscrit. Le lecteur intéressé pourra consulter la première partie de ce mémoire, qui fournit un état de l'art plus complet sur la platitude différentielle, et également sur la détermination de sorties plates par des outils de calcul formel.

Comme nous l'avons souligné dans la première partie, l'intérêt principal du concept de

---

<sup>34</sup>Le contexte n'étant pas ambigu ici, nous désignerons indifféremment la fonctionnelle de coût par fonction objectif, ou encore indice de performance.

platitude réside dans la possibilité de trouver une relation d'équivalence entre la dynamique non linéaire d'un système et une chaîne d'intégrateurs purs, c'est à dire un système trivial. Dans ce cas, les trajectoires du système dans l'espace d'état peuvent être directement obtenues grâce à des relations analytiques, sans intégration, à partir des trajectoires des sorties plates. Un système non linéaire est plat s'il existe un ensemble de variables différentiellement indépendantes (dépendant de l'état, des commandes et d'un nombre fini de leurs dérivées), telles que l'état et les commandes peuvent s'exprimer exclusivement en fonction de ces sorties et d'un nombre fini de leurs dérivées.

Considérons à nouveau le système non linéaire (2.1). On suppose en outre que  $m \leq n$ , ce qui signifie que le système d'équations (2.1) est sous-déterminé.

**Définition 2.1** (Platitude différentielle [1]). *Le système non linéaire (2.1) est différentiellement plat (ou, plus brièvement, plat) si et seulement s'il existe un vecteur  $z$  de dimension  $m$ , dont les éléments sont des fonctions différentiellement indépendantes, de la forme*

$$z = \phi(x, u, \dot{u}, \dots, u^{(\alpha)}),$$

tel que

$$\begin{cases} x = \psi_x(z, \dot{z}, \dots, z^{(\beta-1)}), \\ u = \psi_u(z, \dot{z}, \dots, z^{(\beta)}), \end{cases} \quad (2.5)$$

où  $\psi_x$  et  $\psi_u$  représentent des applications lisses sur la variété  $X$ ,  $z_i^{(k)}$  correspond à la dérivée d'ordre  $k$  de la  $i^{\text{ème}}$  composante de  $z$ , et  $\alpha = (\alpha_1, \dots, \alpha_m)$ ,  $\beta = (\beta_1, \dots, \beta_m)$  sont des  $m$ -uplets finis d'entiers.

Le vecteur  $z \in \mathbb{R}^m$  est appelé sortie plate (ou sortie linéarisante).

### 2.3.3 Problème de commande optimale dans l'espace des sorties plates

L'utilisation de la platitude dans le cadre de l'optimisation de trajectoires consiste alors à transcrire le problème de commande optimale (2.4) dans l'espace des sorties plates, et à optimiser non plus les profils des états et des commandes, mais uniquement les profils des sorties plates. Cette approche a initialement été développée dans [131], et a été appliquée avec succès à un certain nombre de problèmes : vol en formation de micro-satellites [132], évitement d'obstacles [133], planification de trajectoires pour un missile [134], commande prédictive [135] et à horizon fuyant [5], génération de trajectoires pour satellites agiles [136], ...

la figure 2.4 illustre l'approche utilisée. La planification d'une trajectoire permettant d'amener le système (2.1) d'un état  $(x(t_0), u(t_0))$  à un état  $(x(t_f), u(t_f))$  revient à déterminer une trajectoire suffisamment lisse dans l'espace des sorties plates, satisfaisant  $z^{(k)}(t_0)$  et  $z^{(k)}(t_f)$ ,  $k = 1, \dots, \beta$ , et un ensemble de contraintes de trajectoires s'exprimant également en fonction des sorties plates et d'un nombre fini de leurs dérivées. Si le système est différentiellement plat, il n'est alors pas nécessaire de résoudre un problème aux

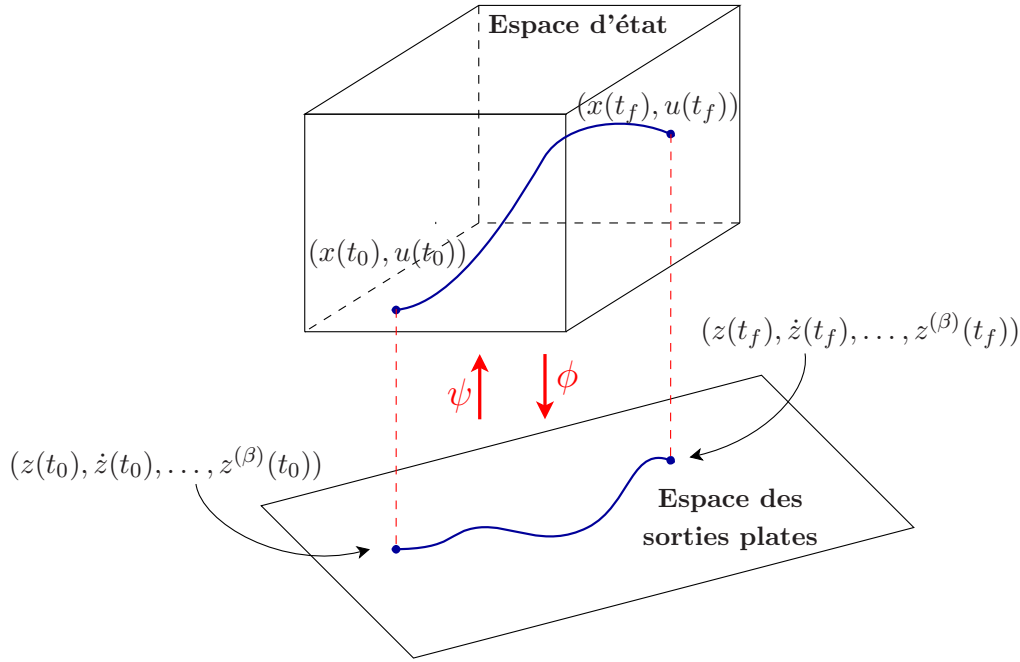


FIG. 2.4 – équivalence entre les trajectoires du système initial et celles du système trivial.

deux bouts. De plus, lorsque les trajectoires des sorties plates sont connues, les relations (2.5) permettent directement de calculer sous forme analytique les profils de l'état et des commandes du système (2.1). Cette propriété constitue un point fort de la stratégie proposée, et sera notamment exploitée dans le cadre du guidage de véhicules de rentrée atmosphérique dans le chapitre 3.

Les trajectoires générées dans l'espace des sorties plates peuvent alors être converties en trajectoires dans l'espace d'état par la relation (2.5) et sans qu'aucune intégration numérique ne soit nécessaire (voir figure 2.4). Une fois la contrainte dynamique (2.1) intervenant dans le problème de commande optimale (2.4) enlevée grâce à la propriété de platitude du système, et en réécrivant la fonctionnelle de coût (2.2) et les contraintes (2.3) en fonction des sorties plates et de leurs dérivées, on obtient un problème de commande optimale équivalent qui s'écrit, dans l'espace des sorties plates, sous la forme

$$\min_{\bar{z}(t)} \quad \mathcal{C}_0(\psi_x(\bar{z}(t_0)), \psi_u(\bar{z}(t_0)), t_0) + \int_{t_0}^{t_f} \mathcal{C}_t(\psi_x(\bar{z}(t)), \psi_u(\bar{z}(t)), t) dt \quad (2.6)$$

$$+ \mathcal{C}_f(\psi_x(\bar{z}(t_f)), \psi_u(\bar{z}(t_f)), t_f)$$

t.q.

$$\begin{aligned} l_0 &\leq A_0 \bar{z}(t_0) &&\leq u_0, \\ l_t &\leq A_t \bar{z}(t) &&\leq u_t, \quad t \in [t_0, t_f], \\ l_f &\leq A_f \bar{z}(t_f) &&\leq u_f, \\ L_0 &\leq c_0(\psi_x(\bar{z}(t_0)), \psi_u(\bar{z}(t_0))) &&\leq U_0, \\ L_t &\leq c_t(\psi_x(\bar{z}(t)), \psi_u(\bar{z}(t))) &&\leq U_t, \quad t \in [t_0, t_f], \\ L_f &\leq c_f(\psi_x(\bar{z}(t_f)), \psi_u(\bar{z}(t_f))) &&\leq U_f. \end{aligned}$$

Les variables de décision du problème de commande optimale sont alors définies par le vecteur des sorties plates et de leurs dérivées, noté  $\bar{z} = (z_1, \dots, z_m, \dot{z}_1, \dots, \dot{z}_m, \dots)$  dans l'espace des jets d'ordre infini<sup>35</sup>, et de dimension  $n_{\bar{z}}$ .

**Remarque 2.4.** Si le système (2.1) n'est pas plat, il est toujours possible d'exprimer les observables autonomes en fonction des variables associées à la paramétrisation de la partie libre<sup>36</sup> du système. Dans ce cas, on obtient une ou plusieurs contraintes de trajectoire supplémentaires de la forme  $\psi_l(v(t)) = 0$ , pour  $t_0 \leq t \leq t_f$ , qu'il faut intégrer au problème de commande optimale (2.6). On obtient ainsi un problème de commande optimale s'exprimant en fonction un nombre réduit de variables d'optimisation. En revanche, si le système (2.1) est (complètement) plat, alors le problème (2.6) comporte un nombre *minimal* de variables d'optimisation.

Nous verrons dans la section 2.6 comment transformer cet OCP de dimension infinie en un problème de programmation non linéaire de dimension finie, en paramétrant les variables d'optimisation, i.e., les sorties plates, par des fonctions candidates (par exemple des B-splines), en approximant la fonctionnelle de coût (2.6) par une loi de quadrature, et en discrétisant le problème de commande optimale sur une échelle de temps donnée.

Mais avant cela, nous allons chercher à répondre à un problème important pouvant survenir lorsque l'on transcrit le problème de commande optimale dans l'espace des sorties plates : il s'agit de la perte de convexité potentielle du problème de commande optimale. En reprenant l'un des points clés du papier de I. M. Ross portant sur le sujet [64], nous pouvons donc soulever le problème suivant : *à quoi cela sert-il de réduire au maximum le nombre de variables d'optimisation si cela conduit à la perte de convexité du problème de commande optimale ?*

Afin de répondre à cette question encore ouverte, nous proposons dans la section suivante une méthodologie originale permettant d'approximer les contraintes de trajectoire du problème de commande optimale par des formes géométriques convexes, appelées superquadriques. Les paramètres de dimensionnement de ces formes sont optimisés par un algorithme génétique, et ces dernières sont ensuite intégrées à l'OCP afin de remplacer les contraintes initiales non convexes. En réalisant une approximation similaire pour la fonctionnelle de coût, on obtient alors un problème de commande optimale convexe. La résolution de ce nouveau problème ne fournit pas seulement une garantie de convergence du solveur utilisé, mais également un gain de temps CPU lors de la planification de trajectoires, le problème de commande optimale n'admettant plus de minimums locaux.

---

<sup>35</sup>remarquons à nouveau qu'il existe une infinité dénombrable de coordonnées  $\bar{z}$  des sorties plates et de leurs dérivées successives dans l'espace des jets d'ordre infini.

<sup>36</sup>c'est à dire sans torsion.

## 2.4 Convexification du problème de commande optimale

Le problème de commande optimale (2.6) obtenu dans la section précédente peut être directement utilisé afin de calculer une trajectoire optimale. Cependant, l'utilisation de la platitude différentielle ne préserve pas en général la convexité<sup>37</sup> du problème de commande optimale initial. Par conséquent, la plupart des contraintes de trajectoire de l'OCP, une fois transcrites dans l'espace des sorties plates, peuvent devenir hautement non linéaires et potentiellement non convexes.

Une publication récente de I. M. Ross [64], traitant de certains problèmes rencontrés lors de la résolution de problèmes de commande optimale, a conclu qu'une diminution successive du nombre de variables d'optimisation, tout en maintenant la convexité du problème, permettrait de gagner en performances (temps CPU) ainsi qu'en robustesse<sup>38</sup>. L'espace des variables d'optimisation résultant de la paramétrisation "plate" du problème de commande optimale étant le plus compact parmi les formulations alternatives étudiées dans [64] (par exemple, les techniques d'inclusion différentielle ou d'inclusion sous forme normale), il semble être très intéressant de pouvoir conserver la formulation plate du problème de manière à obtenir un nombre minimal de variables d'optimisation, tout en assurant que l'OCP reste convexe dans l'espace des sorties plates.

Afin de répondre à ce problème, une première technique d'annexion polyédrique a été proposée dans [12], consistant à élargir un polytope en annexant de plus en plus de portions d'un espace admissible non convexe associé à un ensemble de contraintes non linéaires de trajectoire. Ce problème est posé comme un problème d'optimisation de type semi-infini, dans lequel les variables de décision sont les sommets du polytope, et où la fonctionnelle de coût est donnée par le volume du polytope. Cependant, bien que les polytopes offrent une certaine flexibilité par un large choix de formes géométriques atteignables, ces derniers deviennent particulièrement difficile à manipuler dans les espaces de sorties plates de grande dimension, i.e., au-delà de 3. De plus, l'évaluation de leur volume constitue un problème NP-hard [137]. Récemment, un certain nombre de résultats ont été proposés dans [136], portant sur la convexification de contraintes par des formes superquadriques de dimension  $n$ . Ces travaux constituent le point de départ de notre étude.

Dans cette section, nous proposons une méthodologie originale et efficace permettant de réaliser une approximation intérieure d'un domaine non convexe par une forme géométrique convexe, et différentiable. La procédure consiste à créer une primitive  $\mathcal{S}_I$  à l'intérieur de la région non convexe admissible  $\mathcal{S}_Z$ , et ensuite à la faire grossir jusqu'à ce qu'elle atteigne un volume maximal pour lequel  $\mathcal{S}_I$  se trouve toujours à l'intérieur de

---

<sup>37</sup>nous tenons à faire remarquer ici que les problèmes de commande optimale dans l'espace d'état peuvent aussi bien être convexes que non convexes.

<sup>38</sup>en effet, comme nous l'avons indiqué dans la section 1, les solveurs permettant de résoudre les problèmes de type NLP sont pour la plupart relativement sensibles à l'estimé initial de la trajectoire recherchée.



$S_Z$ . La méthodologie repose sur l'optimisation de formes géométriques déformables, appelées superquadriques, par un algorithme génétique de manière à trouver un minimum global de la fonctionnelle de coût associée au problème d'annexion superellipsoïdale. En outre, nous présentons dans cette section un certain nombre de résultats mathématiques nouveaux portant sur les formes géométriques superquadriques de dimension  $n$ . De plus, l'utilisation d'un ensemble de transformations géométriques  $n$ -D agissant sur les formes superquadriques, et préservant leur convexité, constitue une amélioration importante par rapport aux autres travaux sur le sujet. Les contraintes de trajectoire et la fonctionnelle de coût du problème de commande optimale initial peuvent alors être remplacées par les formes superquadriques optimales correspondantes. Le nouvel OCP est alors résolu dans un sous-espace convexe inclus dans l'espace initial des contraintes. Malgré un certain degré de conservatisme inhérent à l'approche utilisée, la nouvelle formulation convexe du problème de commande optimale engendre à la fois une amélioration de la robustesse du solveur NLP utilisé, et également un gain de temps CPU non négligeable lors du calcul de trajectoires.

L'ensemble de la méthodologie a été implémentée dans une bibliothèque, appelée OCEANS (Optimal Convexification by Evolutionary Algorithms aNd Superquadrics) sous l'environnement Matlab.

### 2.4.1 Formes superquadriques généralisées

Dans les dernières décennies, de nouvelles collections d'objets paramétriques lisses ont été développées dans le cadre de la conception assistée par ordinateur (CAD) et des méthodes numériques pour la création tri-dimensionnelle (3D). D'abord, les *superellipses* ont été proposées afin d'étendre l'ensemble des primitives géométriques traditionnelles, telles que les surfaces quadriques et les nappes paramétriques. Quelques années plus tard, les *superquadriques* ont été introduites dans [138] comme une généralisation en trois dimensions des superellipses, et ont été appliquées à un certain nombre de problèmes. Peut-être la classe la plus générale de primitives géométriques est celle des *hyperquadriques*, obtenues en considérant la somme d'un nombre arbitraires de puissances de termes linéaires [139], et qui comprend les formes quadriques et superquadriques en tant que cas particuliers. Les hyperquadriques sont des formes génératrices dont les polytopes englobants possèdent un nombre arbitraire de facettes. Cependant, même si le nombre de formes candidates atteignables est théoriquement illimité<sup>39</sup>, les hyperquadriques ne possèdent pas de paramétrisation explicite, cette dernière étant obtenue grâce à des méthodes numériques d'optimisation.

**Remarque 2.5.** De nombreux autres types de primitives géométriques existent, telles que les superformes [140], qui sont des formes superquadriques comportant un nombre arbitraire de symétries, les ratioquadriques [141], les cylindres généralisés [142] et les géons [143], etc.

Afin de réaliser un compromis entre la complexité de la représentation de la forme géométrique, i.e., le nombre de formes atteignables, et l'efficacité numérique dans les espaces

<sup>39</sup>comme dans le cas des polytopes possédant un nombre arbitraire de sommets ou de facettes.



de sorties plates d'ordre élevé, nous avons choisi de travailler avec des primitives superquadriques afin de réaliser l'approximation intérieure d'un ensemble atteignable non convexe  $S_Z$ . La classe des formes superquadriques comprend les *superellipsoïdes*, que nous considérons dans la suite, ainsi que les superhyperboloïdes et les supertorroïdes [138]. L'avantage principal est le nombre réduit de paramètres nécessaires pour décrire la forme globale (propriété de compacité de la représentation), et également l'existence d'une paramétrisation explicite de la surface. Cependant, comparé aux primitives hyperquadriques, le nombre de formes candidates est limité et, dans certains cas, des transformations géométriques préservant la convexité peuvent être nécessaires, de manière à réduire le conservatisme de l'approximation.

**Remarque 2.6.** La quantification du conservatisme inhérent à l'approximation superellipsoïdale réalisée représente une donnée importante du problème de commande optimale convexe. En effet, la recherche d'une trajectoire optimale, ou même d'une trajectoire admissible d'un système en fonctionnement défaillant peut conduire à des trajectoires situées sur les frontières des contraintes considérées. En d'autres termes, il se peut que le problème d'optimisation convexe n'admette pas de solution, alors que le problème non convexe initial en admet une. Le degré de conservatisme peut être mesuré par exemple en comparant le volume du domaine admissible initial à celui de la forme géométrique déformable défini dans la proposition 2.7.

Malgré tout, les surfaces superquadriques offrent un degré de flexibilité plus important que les ellipsoïdes par la modification des paramètres de forme dans chaque direction. Plus précisément, les degrés de liberté correspondent aux paramètres de dimensionnement superquadriques, c'est à dire aux demi-grand axes (directions et amplitudes) et aux paramètres de forme. Ainsi, nous allons chercher à étendre en dimension  $n$ , dans le cadre de la méthodologie proposée, un certain nombre de résultats et de transformations obtenus uniquement pour les applications tri-dimensionnelles. Nous proposons ainsi un certain nombre de résultats nouveaux concernant la paramétrisation des formes superquadriques de dimension  $n$ .

Par définition, une *superellipse* en deux dimensions est une courbe fermée du plan  $(O, x, y)$ , décrite par l'équation implicite

$$\left(\frac{x}{a_1}\right)^{\frac{2}{\varepsilon}} + \left(\frac{y}{a_2}\right)^{\frac{2}{\varepsilon}} = 1, \quad (2.7)$$

où  $a_1$  et  $a_2$  représentent les *demi-grand axes*, et  $\varepsilon \in \mathbb{R}^{+,*}$  le *paramètre de forme* de la superquadrique [138].

**Remarque 2.7.** Afin d'évaluer numériquement l'équation (2.7), notons qu'il est nécessaire que les termes  $\frac{x}{a_1}$  et  $\frac{y}{a_2}$  soient d'abord élevés au carré avant d'appliquer la puissance non entière en  $1/\varepsilon$ , et ce afin d'assurer que le résultat du calcul soit bien un nombre réel et non un complexe pour  $x < 0$  ou  $y < 0$ .

En fonction des valeurs des paramètres  $a_1$ ,  $a_2$  et  $\varepsilon$ , il est possible de modifier la forme de la superellipse. Ainsi, pour  $a_1 = a_2$  et  $\varepsilon = 1$ , on obtient le cercle centré en 0 de rayon

$a_1$ . Lorsque  $\varepsilon$  tend vers 0, la superellipse devient un carré dont les cotés sont parallèles aux axes du repère et de longueur unité, et lorsque  $\varepsilon$  devient grand, la superellipse se transforme progressivement en une étoile à quatre branches.

En coordonnées cartésiennes, la paramétrisation explicite associée à l'équation (2.7) s'écrit :

$$\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} a_1 \cos^\varepsilon \theta \\ a_2 \cos^\varepsilon \theta \end{pmatrix}.$$

L'angle  $\theta$ , avec  $-\pi \leq \theta \leq \pi$ , est alors appelé l'*anomalie*. Par convention, les puissances des fonctions trigonométriques considérées sont évaluées comme des fonctions de puissances signées, telles que

$$\begin{cases} \cos^\varepsilon \theta = \text{sign}(\cos \theta) |\cos \theta|^\varepsilon, \\ \sin^\varepsilon \theta = \text{sign}(\sin \theta) |\sin \theta|^\varepsilon. \end{cases}$$

Les superellipsoïdes d'ordres plus élevés s'obtiennent alors en calculant le produit sphérique entre deux superellipses élémentaires de dimension inférieure. Par exemple, un superellipsoïde en dimension 3 est défini par

$$\begin{aligned} \begin{pmatrix} x \\ y \\ z \end{pmatrix} &= \begin{pmatrix} a_{1,1} \cos^{\varepsilon_1} \theta_1 \\ a_{1,2} \sin^{\varepsilon_1} \theta_1 \end{pmatrix} \otimes \begin{pmatrix} a_{2,1} \cos^{\varepsilon_2} \theta_2 \\ a_{2,2} \sin^{\varepsilon_2} \theta_2 \end{pmatrix} \\ &= \begin{pmatrix} a_1 \cos^{\varepsilon_2} \theta_2 \cos^{\varepsilon_1} \theta_1 \\ a_2 \cos^{\varepsilon_2} \theta_2 \sin^{\varepsilon_1} \theta_1 \\ a_3 \sin^{\varepsilon_2} \theta_2 \end{pmatrix}, \end{aligned}$$

avec  $a_1 = a_{1,1}a_{2,1}$ ,  $a_2 = a_{1,1}a_{2,2}$ ,  $a_3 = a_{1,2}$ , et où les anomalies  $\theta_1$  et  $\theta_2$  varient sur les plages  $-\pi \leq \theta_1 < \pi$  et  $-\pi/2 \leq \theta_2 \leq \pi/2$ .

Dans le repère  $(O, x, y, z)$ , le vecteur  $\vec{r} = (x, y, z)^T$  décrivant la surface du superellipsoïde 3D en fonction des paramètres  $a_1, a_2, a_3, \varepsilon_1$  et  $\varepsilon_2$  est représenté sur la figure 2.5. Nous pouvons remarquer sur la figure 2.5 que le paramètre  $\varepsilon_2$  modifie la forme du superellipsoïde sur les plans parallèles à  $(O, x, y)$  et  $\varepsilon_1$  sur les plans perpendiculaires à  $(O, x, y)$ .

Avec les définitions précédentes, il est alors possible de décrire un superellipsoïde de dimension  $n$  en calculant le produit sphérique entre  $n - 1$  superellipses élémentaires définies par

$$E_i(\varepsilon_i, \theta_i) = \begin{pmatrix} a_{i,1} \cos^{\varepsilon_i} \theta_i \\ a_{i,2} \sin^{\varepsilon_i} \theta_i \end{pmatrix}, \quad a_{i,1}, a_{i,2}, \varepsilon_i \in \mathbb{R}^{+,*}, \quad \theta_i \in \begin{cases} [-\pi, \pi[ & \text{si } i = 1 \\ [-\pi/2, \pi/2] & \text{si } i = 2, n - 1 \end{cases},$$

Dans la suite, on désignera par  $\theta = [\theta_1, \dots, \theta_{n-1}]$  le *vecteur des anomalies*, et par  $\varepsilon = [\varepsilon_1, \varepsilon_{n-1}]$  le *vecteur des paramètres de forme*. Les coordonnées  $x = [x_1, \dots, x_n]^T$  d'un point de la surface du superellipsoïde sont alors obtenues en calculant le produit sphérique

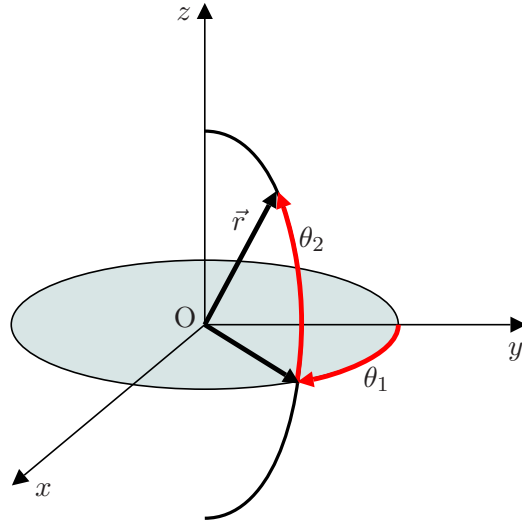


FIG. 2.5 – illustration géométrique des paramètres de dimensionnement d'un superellipsoïde 3D.

$$\bigotimes_{k=1}^{n-1} E_{n-k} = E_{n-1} \otimes E_{n-2} \otimes \dots \otimes E_1, \text{ tel que}$$

$$x_i = \begin{cases} a_1 \prod_{k=1}^{n-1} \cos^{\varepsilon_k} \theta_k, & i = 1, \\ a_i \sin^{\varepsilon_{i-1}} \theta_{i-1} \prod_{k=i}^{n-1} \cos^{\varepsilon_k} \theta_k, & i = 2, \dots, n-1, \\ a_n \sin^{\varepsilon_{n-1}} \theta_{n-1}, & i = n, \end{cases}$$

où chaque demi-grand axe  $a_i$  est obtenu comme une fonction<sup>40</sup> des demi-grand axes élémentaires  $e_{1,1}, e_{1,2}, \dots, e_{n-1,1}, e_{n-1,2}$ . Nous noterons alors  $a = [a_1, \dots, a_n]$  le vecteur des demi-grand axes. En résumé,  $2n - 1$  paramètres sont donc nécessaires pour décrire une superquadrique de dimension  $n$ .

Nous pouvons à présent définir formellement la *paramétrisation trigonométrique explicite* d'un ellipsoïde superquadrique, en coordonnées cartésiennes.

**Proposition 2.1** (paramétrisation trigonométrique  $n$ -D). *Soit  $\mathcal{S}$  un ellipsoïde superquadrique de dimension  $n$ , défini par le vecteur de demi-grand axes  $a \in \mathbb{R}^n$ , le vecteur de paramètres de forme  $\varepsilon \in \mathbb{R}^{n-1}$  et paramétré par le vecteur d'anomalies  $\theta \in \mathbb{R}^{n-1}$ . Alors, la paramétrisation trigonométrique associée, de coordonnées cartésiennes  $x_i, i = 1, \dots, n$ ,*

<sup>40</sup>nous ne chercherons pas à déterminer analytiquement ces fonctions, cela ne présentant pas d'intérêt pratique pour la suite du développement.

est définie par

$$x_i = \begin{cases} a_1 \prod_{k=1}^{n-1} \cos^{\varepsilon_k} \theta_k, & i = 1, \\ a_i \sin^{\varepsilon_{i-1}} \theta_{i-1} \prod_{k=i}^{n-1} \cos^{\varepsilon_k} \theta_k, & i = 2, \dots, n-1, \\ a_n \sin^{\varepsilon_{n-1}} \theta_{n-1}, & i = n, \end{cases} \quad (2.8)$$

où le vecteur d'anomalies  $\theta$  vérifie

$$\begin{aligned} \theta_i &\in [-\pi, \pi[, & i = 1, \\ \theta_i &\in [-\frac{\pi}{2}, \frac{\pi}{2}], & i = 2, \dots, n-1. \end{aligned}$$

Comme nous l'avons mentionné précédemment, les paramètres de forme  $\varepsilon$  permettent de définir la convexité de la forme superquadrique. Notamment, Barr [138] a montré qu'un superellipsoïde 3D est convexe si  $\varepsilon_1, \varepsilon_2 \leq 2$ . Par extension, nous pouvons affirmer qu'un ellipsoïde superquadrique de dimension  $n$  est convexe si,  $\forall i = 1, \dots, n, \varepsilon_i \in ]0, 2]$ . Quelques illustrations de superellipsoïdes en dimension 3 sont données sur la figure 2.6.

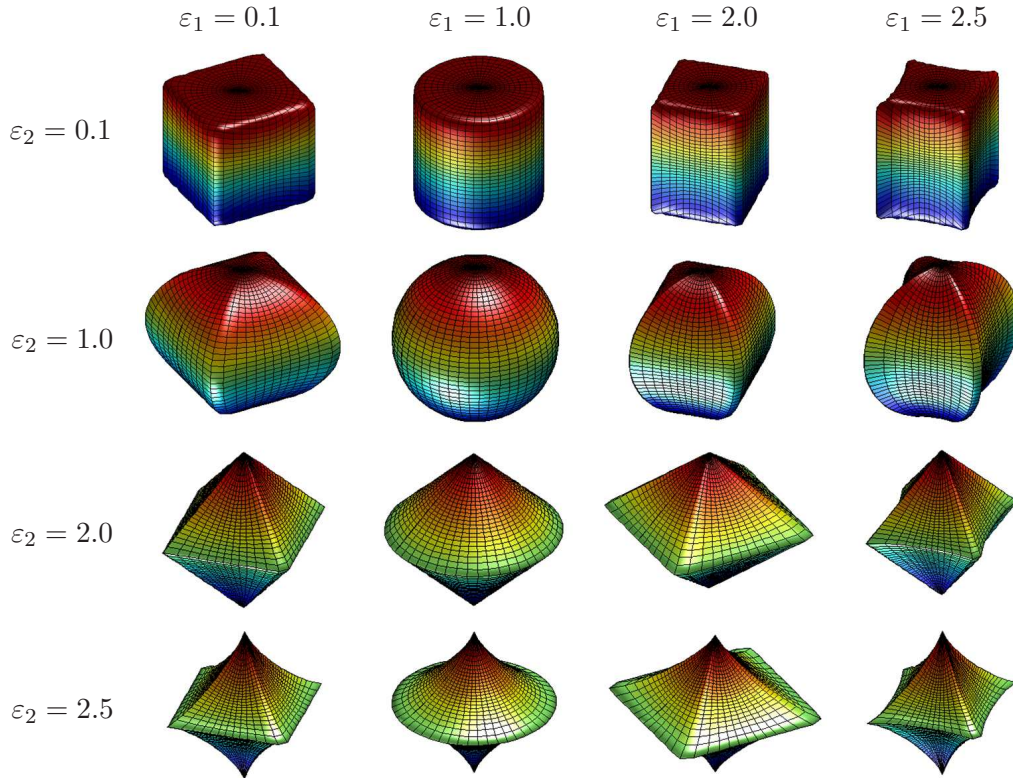


FIG. 2.6 – exemples de formes superquadriques 3D.

La définition suivante introduit la notion de fonction d'appartenance à un superellipsoïde de dimension  $n$ .

**Proposition 2.2** (fonction d'appartenance  $n$ -D). Soit  $\mathcal{S}$  un superellipsoïde de dimension  $n$ , défini par le vecteur de demi-grand axes  $a \in \mathbb{R}^n$ , le vecteur de paramètres de forme  $\varepsilon \in \mathbb{R}^{n-1}$  et paramétré par le vecteur d'anomalies  $\theta \in \mathbb{R}^{n-1}$ . La fonction d'appartenance au superellipsoïde  $n$ -D, associée à la représentation explicite (2.8), est donnée par  $\mathcal{F}_n(x) = \Lambda_{n,n}(x)$ , où  $\Lambda_{n,n}(x)$  vérifie la relation analytique réursive

$$\begin{cases} \Lambda_{n,2}(x) = \left(\frac{x_1}{a_1}\right)^{\frac{2}{\varepsilon_1}} + \left(\frac{x_2}{a_2}\right)^{\frac{2}{\varepsilon_1}}, \\ \Lambda_{n,k}(x) = (\Lambda_{n,k-1}(x))^{\frac{\varepsilon_{k-2}}{\varepsilon_{k-1}}} + \left(\frac{x_k}{a_k}\right)^{\frac{2}{\varepsilon_{k-1}}}, \end{cases} \quad (2.9)$$

où  $x$  représente le vecteur de coordonnées cartésiennes de dimension  $n$ .

**Preuve.** D'après la paramétrisation trigonométrique explicite (2.8), pour un superellipsoïde de dimension  $n$ , les deux premières coordonnées s'écrivent

$$x_1 = a_1 \prod_{i=1}^{n-1} \cos^{\varepsilon_i} \theta_i, \quad (2.10)$$

et

$$x_2 = a_2 \sin^{\varepsilon_1} \theta_1 \prod_{i=2}^{n-1} \cos^{\varepsilon_i} \theta_i. \quad (2.11)$$

En utilisant les expressions (2.10) et (2.11), alors  $\Lambda_{n,2}(x)$  peut se réécrire

$$\Lambda_{n,2}(x) = \left(\prod_{i=1}^{n-1} \cos^{\varepsilon_i} \theta_i\right)^{\frac{2}{\varepsilon_1}} + \left(\sin^{\varepsilon_1} \theta_1 \prod_{i=2}^{n-1} \cos^{\varepsilon_i} \theta_i\right)^{\frac{2}{\varepsilon_1}} \quad (2.12)$$

$$= \cos^2 \theta_1 \prod_{i=2}^{n-1} \cos^{\frac{2\varepsilon_i}{\varepsilon_1}} \theta_i + \sin^2 \theta_1 \prod_{i=2}^{n-1} \cos^{\frac{2\varepsilon_i}{\varepsilon_1}} \theta_i \quad (2.13)$$

$$= \prod_{i=2}^{n-1} \cos^{\frac{2\varepsilon_i}{\varepsilon_1}} \theta_i. \quad (2.14)$$

Par extension à la dimension  $n$ , supposons que  $\Lambda_{n,k}(x)$  s'écrive

$$\Lambda_{n,k}(x) = \prod_{i=k}^{n-1} \cos^{\frac{2\varepsilon_i}{\varepsilon_{k-1}}} \theta_i. \quad (2.15)$$

Alors, comme la coordonnée  $x_{k+1}$  s'écrit

$$x_{k+1} = a_{k+1} \sin^{\varepsilon_k} \theta_k \prod_{i=k+1}^{n-1} \cos^{\varepsilon_i} \theta_i,$$

on obtient

$$\begin{aligned}
 (\Lambda_{n,k}(x))^{\frac{\varepsilon_{k-1}}{\varepsilon_k}} + \left(\frac{x_{k+1}}{a_{k+1}}\right)^{\frac{2}{\varepsilon_k}} &= \left(\prod_{i=k}^{n-1} \cos^{\frac{2\varepsilon_i}{\varepsilon_{k-1}}} \theta_i\right)^{\frac{\varepsilon_{k-1}}{\varepsilon_k}} + \left(\sin^{\varepsilon_k} \theta_k \prod_{i=k+1}^{n-1} \cos^{\varepsilon_i} \theta_i\right)^{\frac{2}{\varepsilon_k}} \\
 &= \prod_{i=k}^{n-1} \cos^{\frac{2\varepsilon_i}{\varepsilon_k}} \theta_i + \sin^2 \theta_k \prod_{i=k+1}^{n-1} \cos^{\frac{2\varepsilon_i}{\varepsilon_k}} \theta_i \\
 &= \cos^2 \theta_k \prod_{i=k+1}^{n-1} \cos^{\frac{2\varepsilon_i}{\varepsilon_k}} \theta_i + \sin^2 \theta_k \prod_{i=k+1}^{n-1} \cos^{\frac{2\varepsilon_i}{\varepsilon_k}} \theta_i \\
 &= \prod_{i=k+1}^{n-1} \cos^{\frac{2\varepsilon_i}{\varepsilon_k}} \theta_i \\
 &= \Lambda_{n,k+1}(x).
 \end{aligned}$$

La relation de récurrence (2.15) est donc vérifiée. En particulier, pour  $k = n$ , on a

$$\Lambda_{n,n}(x) = (\Lambda_{n,n-1}(x))^{\frac{\varepsilon_{n-2}}{\varepsilon_{n-1}}} + \left(\frac{x_n}{a_n}\right)^{\frac{2}{\varepsilon_{n-1}}}.$$

En posant  $x_n = a_n \sin^{\varepsilon_{n-1}} \theta_{n-1}$ , il vient

$$\begin{aligned}
 \Lambda_{n,n}(x) &= \left(\cos^{\frac{2\varepsilon_{n-1}}{\varepsilon_{n-2}}} \theta_{n-1}\right)^{\frac{\varepsilon_{n-2}}{\varepsilon_{n-1}}} + (\sin^{\varepsilon_{n-1}} \theta_{n-1})^{\frac{2}{\varepsilon_{n-1}}} \\
 &= 1,
 \end{aligned}$$

ce qui termine la preuve car, pour un point  $P$  situé à la surface du superellipsoïde, on a

$$\mathcal{F}_n(x) = \Lambda_{n,n}(x) = 1.$$

■

La fonction d'appartenance  $\mathcal{F}_n(x)$  jouit des propriétés suivantes :

- $\mathcal{F}_n(x) < 1$  : le point se trouve à l'intérieur du superellipsoïde,
- $\mathcal{F}_n(x) = 1$  : le point est à la surface du superellipsoïde,
- $\mathcal{F}_n(x) > 1$  : le point se trouve à l'extérieur du superellipsoïde.

Les points obtenus en utilisant la paramétrisation trigonométrique explicite (2.8) ne sont pas régulièrement espacés à la surface du superellipsoïde (voir la figure 2.7). Comme notre objectif sera à terme de développer une primitive à l'intérieur d'un domaine admissible supposé être non convexe, un quadrillage plus uniforme de la surface de la forme géométrique est préférable afin de faciliter le processus d'optimisation.

Dans ce sens, un algorithme de quadrillage paramétrique optimal a été proposé dans [144] pour les superquadriques de petites dimensions, dans lequel un modèle différentiel est utilisé afin de contrôler les propriétés de courbure locale des formes. Cependant, l'algorithme devient coûteux en calcul pour des formes géométriques de dimension plus élevée.

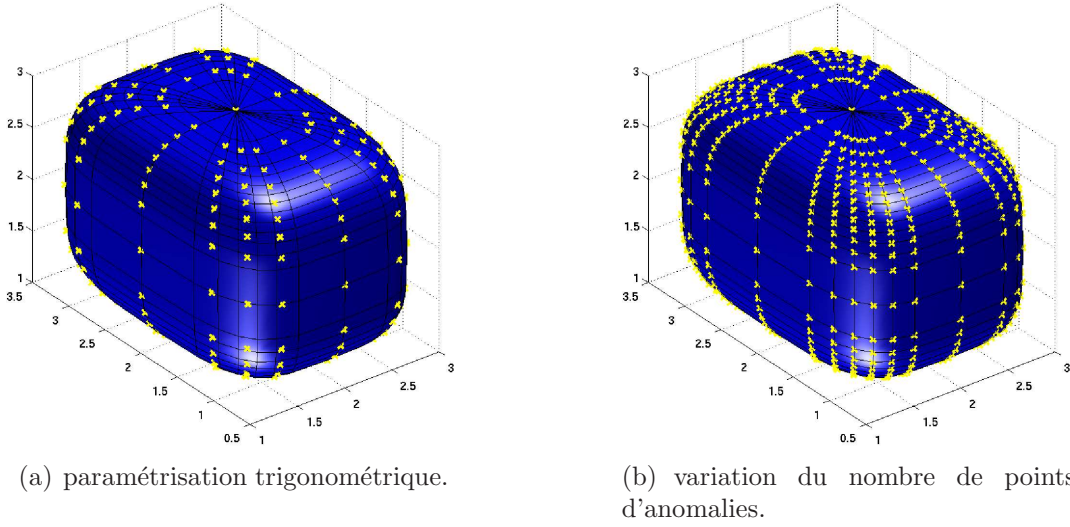


FIG. 2.7 – paramétrisation trigonométrique explicite d'un superellipsoïde 3D.

Dans ce cas, une technique de quadrillage paramétrique sous-optimale peut être utilisée. Ainsi, nous considérons ici l'extension en  $n$  dimensions de la représentation "angle-centre" décrite dans [145].

**Proposition 2.3** (paramétrisation angle-centre  $n$ -D). *Soit  $\mathcal{S}$  un superellipsoïde de dimension  $n$ , défini par le vecteur de demi-grand axes  $a \in \mathbb{R}^n$ , le vecteur de paramètres de forme  $\varepsilon \in \mathbb{R}^{n-1}$ , et paramétré par le vecteur d'anomalies  $\theta \in \mathbb{R}^{n-1}$ . Alors, la paramétrisation angle-centre associée, de coordonnées cartésiennes  $x_i$ ,  $i = 1, \dots, n$ , est définie par*

$$x_i = \begin{cases} r(\theta) \prod_{k=1}^{n-1} \cos \theta_k, & i = 1, \\ r(\theta) \sin \theta_{i-1} \prod_{k=i}^{n-1} \cos \theta_k, & i = 2, \dots, n-1, \\ r(\theta) \sin \theta_{n-1}, & i = n, \end{cases} \quad (2.16)$$

où le rayon  $r(\theta) = \frac{1}{\chi_{n,n}}$  est calculé au moyen de la récursion suivante :

$$\begin{cases} \chi_{n,2} = \left[ \left( \frac{\prod_{k=1}^{n-1} \cos \theta_k}{a_1} \right)^{\frac{2}{\varepsilon_1}} + \left( \frac{\sin \theta_1 \prod_{k=2}^{n-1} \cos \theta_k}{a_2} \right)^{\frac{2}{\varepsilon_1}} \right]^{\frac{\varepsilon_1}{2}}, & j = 2, \\ \chi_{n,j} = \left[ (\chi_{n,j-1})^{\frac{2}{\varepsilon_{j-1}}} + \left( \frac{\sin \theta_{j-1} \prod_{k=j}^{n-1} \cos \theta_k}{a_j} \right)^{\frac{2}{\varepsilon_{j-1}}} \right]^{\frac{\varepsilon_{j-1}}{2}}, & j = 3, \dots, n, \end{cases} \quad (2.17)$$

avec

$$\begin{aligned} \theta_i &\in [-\pi, \pi[, & i = 1, \\ \theta_i &\in [-\frac{\pi}{2}, \frac{\pi}{2}], & i = 2, \dots, n-1. \end{aligned}$$

**Preuve.** Cherchons à déterminer le lien existant entre les coordonnées (2.16) de la paramétrisation angle-centre, et les coordonnées (2.8) de la paramétrisation trigonométrique. En identifiant ces deux expressions terme à terme, il vient

$$\begin{cases} r(\theta) \prod_{k=1}^{n-1} \cos \theta_k = a_1 \prod_{k=1}^{n-1} \cos^{\varepsilon_k} \theta_k, & i = 1, \\ r(\theta) \sin \theta_{i-1} \prod_{k=i}^{n-1} \cos \theta_k = a_i \sin^{\varepsilon_{i-1}} \theta_{i-1} \prod_{k=i}^{n-1} \cos^{\varepsilon_k} \theta_k, & i = 2, \dots, n-1, \\ r(\theta) \sin \theta_{n-1} = a_n \sin^{\varepsilon_{n-1}} \theta_{n-1}, & i = n. \end{cases}$$

Pour  $i = 1$ , à partir de la relation

$$r(\theta) \prod_{k=1}^{n-1} \cos \theta_k = a_1 \prod_{k=1}^{n-1} \cos^{\varepsilon_k} \theta_k,$$

on obtient

$$\begin{aligned} \left( \frac{\prod_{k=1}^{n-1} \cos \theta_k}{a_1} \right)^{\frac{2}{\varepsilon_1}} &= \left( \frac{\prod_{k=1}^{n-1} \cos^{\varepsilon_k} \theta_k}{r(\theta)} \right)^{\frac{2}{\varepsilon_1}} \\ &= \frac{\cos^2 \theta_1}{r(\theta)^{\frac{2}{\varepsilon_1}}} \prod_{k=2}^{n-1} \cos^{\frac{2\varepsilon_k}{\varepsilon_1}} \theta_k. \end{aligned}$$

Ensuite, pour  $i = 2$ , on obtient, à partir de

$$r(\theta) \sin \theta_1 \prod_{k=2}^{n-1} \cos \theta_k = a_2 \sin^{\varepsilon_1} \theta_1 \prod_{k=2}^{n-1} \cos^{\varepsilon_k} \theta_k,$$

l'expression suivante :

$$\begin{aligned} \left( \frac{\sin \theta_1 \prod_{k=2}^{n-1} \cos \theta_k}{a_2} \right)^{\frac{2}{\varepsilon_1}} &= \left( \frac{\sin^{\varepsilon_1} \theta_1 \prod_{k=2}^{n-1} \cos^{\varepsilon_k} \theta_k}{r(\theta)} \right)^{\frac{2}{\varepsilon_1}} \\ &= \frac{\sin^2 \theta_1}{r(\theta)^{\frac{2}{\varepsilon_1}}} \prod_{k=2}^{n-1} \cos^{\frac{2\varepsilon_k}{\varepsilon_1}} \theta_k. \end{aligned}$$



En regroupant les expressions précédentes, on obtient

$$\begin{aligned}
 \chi_{n,2} &= \left[ \left( \frac{\prod_{k=1}^{n-1} \cos \theta_k}{a_1} \right)^{\frac{2}{\varepsilon_1}} + \left( \frac{\sin \theta_1 \prod_{k=2}^{n-1} \cos \theta_k}{a_2} \right)^{\frac{2}{\varepsilon_1}} \right]^{\frac{\varepsilon_1}{2}} \\
 &= \left[ \left( \frac{\cos^2 \theta_1}{r(\theta)^{\frac{2}{\varepsilon_1}}} + \frac{\sin^2 \theta_1}{r(\theta)^{\frac{2}{\varepsilon_1}}} \right) \prod_{k=2}^{n-1} \cos^{\frac{2\varepsilon_k}{\varepsilon_1}} \theta_k \right]^{\frac{\varepsilon_1}{2}} \\
 &= \frac{1}{r(\theta)} \prod_{k=2}^{n-1} \cos^{\varepsilon_k} \theta_k.
 \end{aligned}$$

Par extension à la dimension  $n$ , supposons que  $\chi_{n,j-1}$  puisse s'écrire

$$\chi_{n,j-1} = \frac{1}{r(\theta)} \prod_{k=j-1}^{n-1} \cos^{\varepsilon_k} \theta_k, \quad (2.18)$$

et que

$$\left( \frac{\sin \theta_{j-1} \prod_{k=j}^{n-1} \cos \theta_k}{a_j} \right)^{\frac{2}{\varepsilon_{j-1}}} = \frac{\sin^2 \theta_{j-1}}{r(\theta)^{\frac{2}{\varepsilon_{j-1}}}} \prod_{k=j}^{n-1} \cos^{\frac{2\varepsilon_k}{\varepsilon_{j-1}}} \theta_k. \quad (2.19)$$

De plus, supposons que la relation de récurrence

$$\chi_{n,j} = \left[ (\chi_{n,j-1})^{\frac{2}{\varepsilon_{j-1}}} + \left( \frac{\sin \theta_{j-1} \prod_{k=j}^{n-1} \cos \theta_k}{a_j} \right)^{\frac{2}{\varepsilon_{j-1}}} \right]^{\frac{\varepsilon_{j-1}}{2}} \quad (2.20)$$

soit vraie.

Alors, en combinant (2.18), (2.19) et (2.20),  $\chi_{n,j}$  peut se réécrire sous la forme

$$\begin{aligned}
 \chi_{n,j} &= \left[ \left( \frac{1}{r(\theta)} \prod_{k=j-1}^{n-1} \cos^{\varepsilon_k} \theta_k \right)^{\frac{2}{\varepsilon_{j-1}}} + \left( \frac{\sin \theta_{j-1} \prod_{k=j}^{n-1} \cos \theta_k}{a_j} \right)^{\frac{2}{\varepsilon_{j-1}}} \right]^{\frac{\varepsilon_{j-1}}{2}} \\
 &= \left( \frac{1}{r(\theta)^{\frac{2}{\varepsilon_{j-1}}}} \prod_{k=j-1}^{n-1} \cos^{\frac{2\varepsilon_k}{\varepsilon_{j-1}}} \theta_k + \frac{\sin^2 \theta_{j-1}}{r(\theta)^{\frac{2}{\varepsilon_{j-1}}}} \prod_{k=j}^{n-1} \cos^{\frac{2\varepsilon_k}{\varepsilon_{j-1}}} \theta_k \right)^{\frac{\varepsilon_{j-1}}{2}} \\
 &= \left( \frac{\cos^2 \theta_{j-1}}{r(\theta)^{\frac{2}{\varepsilon_{j-1}}}} \prod_{k=j}^{n-1} \cos^{\frac{2\varepsilon_k}{\varepsilon_{j-1}}} \theta_k + \frac{\sin^2 \theta_{j-1}}{r(\theta)^{\frac{2}{\varepsilon_{j-1}}}} \prod_{k=j}^{n-1} \cos^{\frac{2\varepsilon_k}{\varepsilon_{j-1}}} \theta_k \right)^{\frac{\varepsilon_{j-1}}{2}} \\
 &= \frac{1}{r(\theta)} \prod_{k=j}^{n-1} \cos^{\varepsilon_k} \theta_k,
 \end{aligned}$$

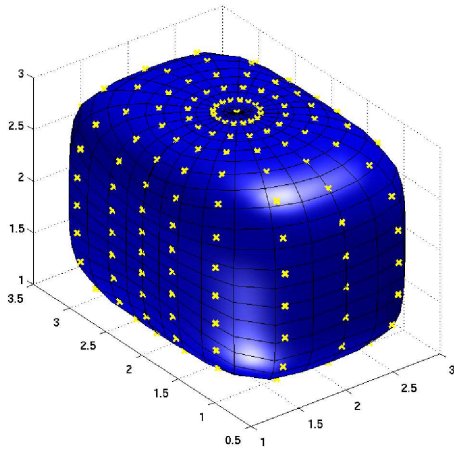
ce qui prouve que (2.18) et (2.19) sont vraies, et donc (2.20) également.

En particulier, pour  $j = n$ , on a

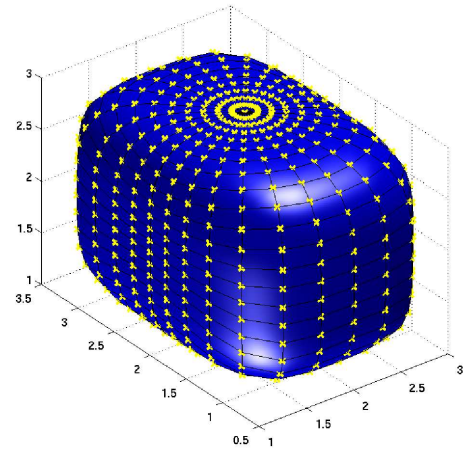
$$\begin{aligned}
 \chi_{n,n} &= \left[ (\chi_{n,n-1})^{\frac{2}{\varepsilon_{n-1}}} + \left( \frac{\sin \theta_{n-1}}{a_n} \right)^{\frac{2}{\varepsilon_{n-1}}} \right]^{\frac{\varepsilon_{n-1}}{2}} \\
 &= \left[ \left( \frac{1}{r(\theta)} \cos^{\varepsilon_{n-1}} \theta_{n-1} \right)^{\frac{2}{\varepsilon_{n-1}}} + \frac{\sin^2 \theta_{n-1}}{r(\theta)^{\frac{2}{\varepsilon_{n-1}}}} \right]^{\frac{\varepsilon_{n-1}}{2}} \\
 &= \left( \frac{\cos^2 \theta_{n-1}}{r(\theta)^{\frac{2}{\varepsilon_{n-1}}} + \frac{\sin^2 \theta_{n-1}}{r(\theta)^{\frac{2}{\varepsilon_{n-1}}}} \right)^{\frac{\varepsilon_{n-1}}{2}} \\
 &= \frac{1}{r(\theta)},
 \end{aligned}$$

d'où  $r(\theta) = \frac{1}{\chi_{n,n}}$ , ce qui termine la preuve. ■

Comme illustré sur la figure 2.8, on obtient une meilleure distribution des points à la surface du superellipsoïde. Ainsi, en choisissant une densité de points appropriée, c'est à dire un nombre suffisant de points d'anomalies, le quadrillage sous-optimal engendré permettra d'éviter que des points du domaine non convexe admissible que l'on cherche à approximer entrent accidentellement à l'intérieur de la forme convexe lors de son expansion.



(a) paramétrisation angle-centre.



(b) variation du nombre de points d'anomalies.

FIG. 2.8 – paramétrisation angle-centre explicite d'un superellipsoïde 3D.

Un autre aspect important lié à l'utilisation de formes superquadriques concerne le nombre de formes atteignables, qui est limité du fait de leur faible nombre de degrés de liberté. Dans la suite, des transformations géométriques permettant de préserver la convexité de la

forme sont introduites afin que l'on puisse approximer un plus grand nombre de contraintes non linéaires non convexes. Ainsi, nous allons considérer la rotation, la translation, et le pincement linéaire le long d'un axe en dimension  $n$ .

**Définition 2.2** (rotation  $n$ -D [146]). Soient  $[x]_b = (x_1, x_2, \dots, x_n)$  les coordonnées cartésiennes d'un superellipsoïde  $\mathcal{S}$  de dimension  $n$ , exprimées dans un repère local centré  $\mathcal{R}_b$ . Sous l'action d'une rotation  $n$ -D, les nouvelles coordonnées superellipsoidales  $[x]_n = (x'_1, x'_2, \dots, x'_n)$ , exprimées dans un système de coordonnées globales  $\mathcal{R}_n$ , sont données par  $[x]_n = R[x]_b$ , où la matrice de rotation  $n$ -D  $R$  est définie de manière récursive par

$$R = \prod_{k=1}^m Q_k(P_k, \Phi_k) = \sum_{k=1}^m Q_k(P_k, \Phi_k) - (m-1)I_n,$$

où  $Q_k$  représente une matrice de rotation élémentaire définie par

$$Q_k(P_k, \Phi_k) = I + (\cos \Phi_k - 1) P_k P_k^T + P_k J P_k^T \sin \Phi_k,$$

où  $P_k$  est une matrice de permutation orthogonale, et  $J$  est une matrice de taille  $2 \times 2$ , appelée matrice symplectique, telle que

$$J = \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix}.$$

En outre, pour un superellipsoïde de dimension  $n$ , le nombre de rotations élémentaires admissibles  $m$  est fini et vaut  $m = \frac{n(n-1)}{2}$ .

La figure 2.9 illustre l'action d'une rotation de  $45^\circ$  sur l'axe  $x$ , et de  $-45^\circ$  sur les axes  $y$  et  $z$ , sur une superquadrique tri-dimensionnelle.

Une autre transformation géométrique courante, peut être la plus simple, est la translation en  $n$  dimensions, rappelée dans la définition suivante.

**Définition 2.3** (translation  $n$ -D [147]). Soient  $[x]_b = (x_1, x_2, \dots, x_n)$  les coordonnées cartésiennes d'un superellipsoïde  $\mathcal{S}$  de dimension  $n$ , exprimées dans un repère local centré  $\mathcal{R}_b$ . Alors, un point de coordonnées  $[x]_b$ , translaté d'une distance  $d = (d_1, d_2, \dots, d_n)$ , admet pour nouvelles coordonnées  $[x]_n = (x'_1, x'_2, \dots, x'_n)$ , qui peuvent être obtenues par  $[x]_n = T[x]_b$ , où, sous forme matricielle étendue et en coordonnées homogènes :

$$\begin{pmatrix} x'_1 \\ x'_2 \\ \vdots \\ x'_n \end{pmatrix} = \begin{pmatrix} 1 & 0 & \cdots & 0 & d_1 \\ 0 & \ddots & \ddots & \vdots & \vdots \\ \vdots & \ddots & \ddots & 0 & \vdots \\ \vdots & & \ddots & \ddots & d_n \\ 0 & \cdots & \cdots & 0 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix}. \quad (2.21)$$

Finalement, la dernière transformation géométrique préservant la convexité de la forme géométrique, que nous considérons dans cette étude, est le *pincement linéaire* le long d'une direction donnée. Ce type de transformation peut être utilisé, par exemple, pour générer des objets non symétriques possédant une forme conique ou prismatique [148].

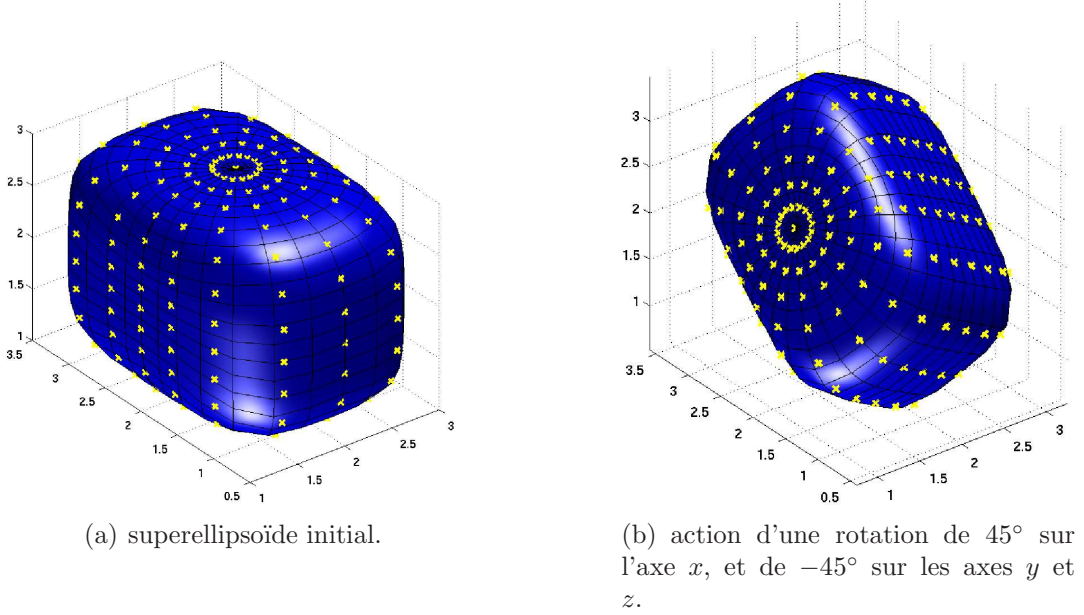


FIG. 2.9 – effet d'une rotation sur une forme superquadrique 3D.

**Définition 2.4** (pincement linéaire  $n$ -D [148]). Soient  $[x]_b = (x_1, x_2, \dots, x_n)$  les coordonnées cartésiennes d'un superellipsoïde  $\mathcal{S}$  de dimension  $n$ . Alors, l'action d'un pincement linéaire, le long d'une direction de pincement  $p$ , engendre une dilatation ou une contraction linéaire des  $(n-1)$  axes restant, selon l'amplitude du vecteur des coefficients de pincement  $v = (v_1, v_2, \dots, v_{n-1})$ . Les coordonnées du superellipsoïde résultant  $[x]_n = (x'_1, x'_2, \dots, x'_n)$  peuvent être calculées en utilisant la relation  $[x]_n = P[x]_b$ , ou, sous forme matricielle,

$$\begin{pmatrix} x'_1 \\ x'_2 \\ \vdots \\ x'_p \\ \vdots \\ x'_n \end{pmatrix} = \begin{pmatrix} \frac{v_1}{a_p} + 1 & 0 & \cdots & \cdots & \cdots & 0 \\ 0 & \frac{v_2}{a_p} + 1 & \ddots & & & \vdots \\ \vdots & & \ddots & 1 & \ddots & \vdots \\ \vdots & & & \ddots & \ddots & 0 \\ 0 & \cdots & \cdots & \cdots & 0 & \frac{v_n}{a_p} + 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_p \\ \vdots \\ x_n \end{pmatrix}. \quad (2.22)$$

L'effet d'un pincement linéaire le long de l'axe vertical d'un superellipsoïde en trois dimensions est illustré sur la figure 2.10. En considérant les trois transformations géométriques précédentes, le vecteur  $\Psi$ , contenant l'ensemble des paramètres de dimensionnement permettant d'obtenir un superellipsoïde positionné, orienté et pincé, est défini par

$$\Psi = \left\{ \underbrace{a_1, \dots, a_n}_{\text{demi-grand axes}}, \underbrace{\varepsilon_1, \dots, \varepsilon_{n-1}}_{\text{par. de forme}}, \underbrace{\Phi_1, \dots, \Phi_{n(n+1)/2}}_{\text{par. de rotation}}, \underbrace{d_1, \dots, d_n}_{\text{par. de translation}}, \underbrace{v_1, \dots, v_{n-1}}_{\text{par. de pincement}} \right\}. \quad (2.23)$$

Les paramétrisations trigonométriques et angle-centre doivent être modifiées afin de tenir compte de ces transformations.

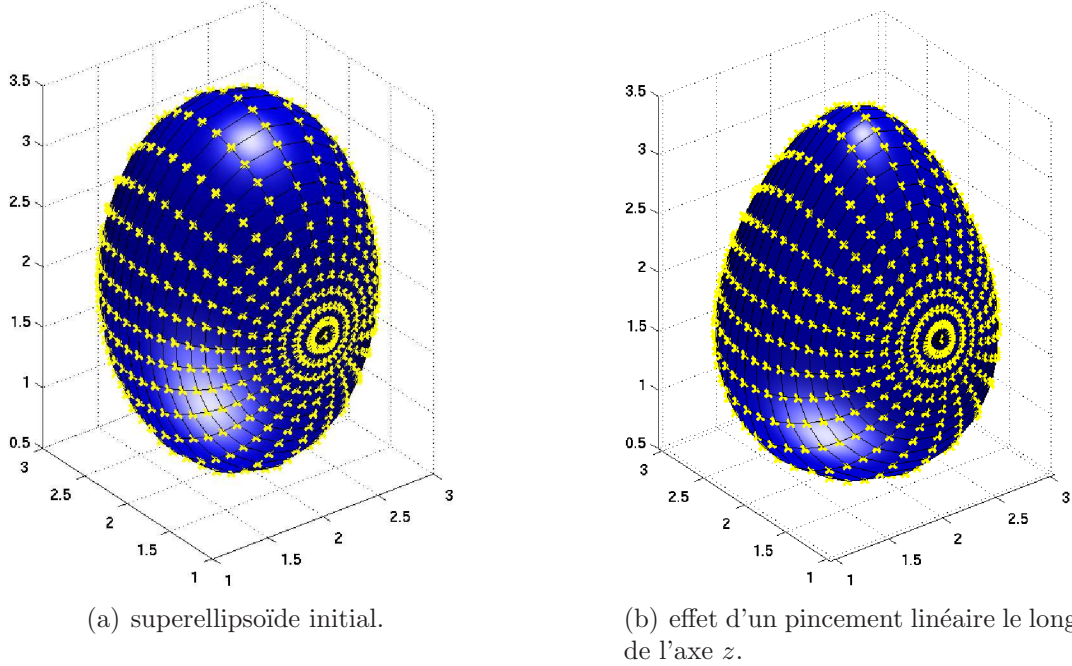


FIG. 2.10 – effet d'un pincement linéaire sur une forme superquadrique 3D.

**Proposition 2.4** (paramétrisation trigonométrique d'un superellipsoïde  $n$ -D déformé). Soit  $\mathcal{S}$  un ellipsoïde superquadrique de dimension  $n$ , décrit par le vecteur  $\Psi$  défini par (2.23). Alors, la paramétrisation trigonométrique associée, de coordonnées cartésiennes  $x_i$ ,  $i = 1, \dots, n$ , est définie par

$$x_i = \begin{cases} a_1(v_1 \sin^{\varepsilon_{p-1}} \theta_{p-1} \prod_{j=p}^{n-1} \cos^{\varepsilon_j} \theta_j + 1) \prod_{k=1}^{n-1} \cos^{\varepsilon_k} \theta_k, & i = 1, \\ a_i(v_i \sin^{\varepsilon_{p-1}} \theta_{p-1} \prod_{j=p}^{n-1} \cos^{\varepsilon_j} \theta_j + 1) \sin^{\varepsilon_{i-1}} \theta_{i-1} \prod_{k=i}^{n-1} \cos^{\varepsilon_k} \theta_k, & i = 2, \dots, n-1, \quad i \neq p, \\ a_p \sin^{\varepsilon_{p-1}} \theta_{p-1} \prod_{j=p}^{n-1} \cos^{\varepsilon_j} \theta_j, & i = p, \\ a_n(v_n \sin^{\varepsilon_{p-1}} \theta_{p-1} \prod_{j=p}^{n-1} \cos^{\varepsilon_j} \theta_j + 1) \sin^{\varepsilon_{n-1}} \theta_{n-1}, & i = n, \end{cases} \quad (2.24)$$

où  $p$  est la direction de pincement ( $v_p = 0$ ). De plus, le vecteur d'anomalies  $\theta$  vérifie

$$\begin{aligned} \theta_i &\in [-\pi, \pi[, & i = 1, \\ \theta_i &\in [-\frac{\pi}{2}, \frac{\pi}{2}], & i = 2, \dots, n-1. \end{aligned}$$

**Preuve.** La paramétrisation (2.24) s'obtient immédiatement en appliquant la matrice de pincement (2.22) aux coordonnées trigonométriques initiales définies par (2.8). ■

De la même manière, la paramétrisation angle-centre explicite (2.16) peut être réécrite afin de tenir compte des transformations géométriques précédentes.

**Proposition 2.5** (paramétrisation angle-centre d'un superellipsoïde  $n$ -D déformé). *Soit  $\mathcal{S}$  un ellipsoïde superquadrique de dimension  $n$ , décrit par le vecteur  $\Psi$  défini par (2.23). Alors, la paramétrisation angle-centre associée, de coordonnées cartésiennes  $x_i$ ,  $i = 1, \dots, n$ , est définie par :*

$$x_i = \begin{cases} r(\theta) \left( \frac{v_1}{a_p} r(\theta) \sin \theta_{p-1} \prod_{j=p}^{n-1} \cos \theta_j + 1 \right) \prod_{k=1}^{n-1} \cos \theta_k, & i = 1, \\ r(\theta) \left( \frac{v_i}{a_p} r(\theta) \sin \theta_{p-1} \prod_{j=p}^{n-1} \cos \theta_j + 1 \right) \sin \theta_{i-1} \prod_{k=i}^{n-1} \cos \theta_k, & i = 2, \dots, n-1, i \neq p, \\ r(\theta) \sin \theta_{p-1} \prod_{j=p}^{n-1} \cos \theta_j, & i = p, \\ r(\theta) \left( \frac{v_n}{a_p} r(\theta) \sin \theta_{p-1} \prod_{j=p}^{n-1} \cos \theta_j + 1 \right) \sin \theta_{n-1}, & i = n, \end{cases} \quad (2.25)$$

où  $p$  est la direction de pincement ( $v_p = 0$ ). Le rayon  $r(\theta) = \frac{1}{\chi_{n,n}}$  est calculé au moyen de la récursion suivante :

$$\begin{cases} \chi_{n,2} = \left[ \left( \frac{\prod_{k=1}^{n-1} \cos \theta_k}{a_1} \right)^{\frac{2}{\varepsilon_1}} + \left( \frac{\sin \theta_1 \prod_{k=2}^{n-1} \cos \theta_k}{a_2} \right)^{\frac{2}{\varepsilon_1}} \right]^{\frac{\varepsilon_1}{2}}, & j = 2, \\ \chi_{n,j} = \left[ (\chi_{n,j-1})^{\frac{2}{\varepsilon_{j-1}}} + \left( \frac{\sin \theta_{j-1} \prod_{k=j}^{n-1} \cos \theta_k}{a_j} \right)^{\frac{2}{\varepsilon_{j-1}}} \right]^{\frac{\varepsilon_{j-1}}{2}}, & j = 3, \dots, n, \end{cases}$$

avec

$$\begin{aligned} \theta_i &\in [-\pi, \pi[, & i = 1, \\ \theta_i &\in [-\frac{\pi}{2}, \frac{\pi}{2}], & i = 2, \dots, n-1. \end{aligned}$$

**Preuve.** En appliquant la matrice de pincement (2.22) aux coordonnées angle-centre définies par (2.16), on obtient immédiatement la paramétrisation (2.25). ■

Finalement, la fonction d'appartenance au superquadrique, définie par (2.9), a été modifiée afin de tenir compte des transformations géométriques précédentes.

**Proposition 2.6** (fonction d'appartenance à un superellipsoïde  $n$ -D déformé). *Soit un superellipsoïde  $\mathcal{S}$  de dimension  $n$ , décrit par le vecteur  $\Psi$  défini par (2.23). Alors, la*

fonction d'appartenance (*implicite*)  $\mathcal{F}_n(\Psi, x) = \Lambda_{n,n}(\Psi, x)$ , est définie par l'expression réursive suivante :

$$\begin{cases} \Lambda_{n,2}(\Psi, x) = \left( \frac{x_1}{a_1 \left( \frac{v_1}{a_p} x_p + 1 \right)} \right)^{\frac{2}{\varepsilon_1}} + \left( \frac{x_2}{a_2 \left( \frac{v_2}{a_p} x_p + 1 \right)} \right)^{\frac{2}{\varepsilon_1}}, \\ \Lambda_{n,k}(\Psi, x) = (\Lambda_{n,k-1}(\Psi, x))^{\frac{\varepsilon_{k-2}}{\varepsilon_{k-1}}} + \left( \frac{x_k}{a_k \left( \frac{v_k}{a_p} x_p + 1 \right)} \right)^{\frac{2}{\varepsilon_{k-1}}}, \end{cases} \quad (2.26)$$

où  $v_p = 0$  dans la direction de pincement  $p$ .

**Preuve.** D'après la paramétrisation trigonométrique explicite (2.24) associée à une superquadrique déformée, les deux premières coordonnées  $x_1$  et  $x_2$  s'écrivent

$$x_1 = a_1 \left( \frac{v_1}{a_p} x_p + 1 \right) \prod_{i=1}^{n-1} \cos^{\varepsilon_i} \theta_i, \quad (2.27)$$

et

$$x_2 = a_2 \left( \frac{v_2}{a_p} x_p + 1 \right) \sin^{\varepsilon_1} \theta_1 \prod_{i=2}^{n-1} \cos^{\varepsilon_i} \theta_i. \quad (2.28)$$

Maintenant, en utilisant (2.27) et (2.28),  $\Lambda_{n,2}(\Psi, x)$  peut se réécrire

$$\begin{aligned} \Lambda_{n,2}(\Psi, x) &= \left( \prod_{i=1}^{n-1} \cos^{\varepsilon_i} \theta_i \right)^{\frac{2}{\varepsilon_1}} + \left( \sin^{\varepsilon_1} \theta_1 \prod_{i=2}^{n-1} \cos^{\varepsilon_i} \theta_i \right)^{\frac{2}{\varepsilon_1}} \\ &= \left( \cos^{\varepsilon_1} \theta_1 \prod_{i=2}^{n-1} \cos^{\varepsilon_i} \theta_i \right)^{\frac{2}{\varepsilon_1}} + \left( \sin^{\varepsilon_1} \theta_1 \prod_{i=2}^{n-1} \cos^{\varepsilon_i} \theta_i \right)^{\frac{2}{\varepsilon_1}} \\ &= \prod_{i=2}^{n-1} \cos^{\frac{2\varepsilon_i}{\varepsilon_1}} \theta_i. \end{aligned}$$

Par extension, pour les superellipsoïdes de dimension  $n$ , supposons que  $\Lambda_{n,k}(\Psi, x)$  s'écrive

$$\Lambda_{n,k}(\Psi, x) = \prod_{i=k}^{n-1} \cos^{\frac{2\varepsilon_i}{\varepsilon_{k-1}}} \theta_i. \quad (2.29)$$

Alors, comme la coordonnée  $x_{k+1}$  s'écrit

$$x_{k+1} = a_{k+1} \left( \frac{v_{k+1}}{a_p} x_p + 1 \right) \sin^{\varepsilon_k} \theta_k \prod_{i=k+1}^{n-1} \cos^{\varepsilon_i} \theta_i,$$

il vient :

$$\begin{aligned}
 & (\Lambda_{n,k}(\Psi, x))^{\frac{\varepsilon_{k-1}}{\varepsilon_k}} + \left( \frac{x_{k+1}}{a_{k+1} \left( \frac{v_{k+1}}{a_p} x_p + 1 \right)} \right)^{\frac{2}{\varepsilon_k}} \\
 &= \left( \prod_{i=k}^{n-1} \cos^{\frac{2\varepsilon_i}{\varepsilon_{k-1}}} \theta_i \right)^{\frac{\varepsilon_{k-1}}{\varepsilon_k}} + \left( \sin^{\varepsilon_k} \theta_k \prod_{i=k+1}^{n-1} \cos^{\varepsilon_i} \theta_i \right)^{\frac{2}{\varepsilon_k}} \\
 &= \cos^2 \theta_k \prod_{i=k+1}^{n-1} \cos^{\frac{2\varepsilon_i}{\varepsilon_k}} \theta_i + \sin^2 \theta_k \prod_{i=k+1}^{n-1} \cos^{\frac{2\varepsilon_i}{\varepsilon_k}} \theta_i \\
 &= \prod_{i=k+1}^{n-1} \cos^{\frac{2\varepsilon_i}{\varepsilon_k}} \theta_i \\
 &= \Lambda_{n,k+1}(\Psi, x).
 \end{aligned}$$

la relation de récurrence (2.29) est donc vérifiée. En particulier, pour  $k = n$ , on a

$$\Lambda_{n,n}(\Psi, x) = (\Lambda_{n,n-1}(\Psi, x))^{\frac{\varepsilon_{n-2}}{\varepsilon_{n-1}}} + \left( \frac{x_n}{a_n} \right)^{\frac{2}{\varepsilon_{n-1}}}.$$

Finalement, en posant

$$x_n = a_n \left( \frac{v_n}{a_p} x_p + 1 \right) \sin^{\varepsilon_{n-1}} \theta_{n-1},$$

on obtient

$$\begin{aligned}
 \Lambda_{n,n}(\Psi, x) &= \left( \cos^{\frac{2\varepsilon_{n-1}}{\varepsilon_{n-2}}} \right)^{\frac{\varepsilon_{n-2}}{\varepsilon_{n-1}}} + (\sin^{\varepsilon_{n-1}} \theta_{n-1})^{\frac{2}{\varepsilon_{n-1}}} \\
 &= \cos^2 \theta_{n-1} + \sin^2 \theta_{n-1} \\
 &= 1.
 \end{aligned}$$

Or, par définition de la fonction d'appartenance, un point  $P$  situé à la surface du superellipsoïde vérifie

$$\mathcal{F}_n(\Psi, x) = \Lambda_{n,n}(\Psi, x) = 1,$$

ce qui termine la preuve. ■

De plus, le volume d'un superellipsoïde de dimension  $n$  peut être obtenu à partir des paramétrisations trigonométriques ou angle-centre, définies respectivement par (2.24) et (2.25). Ce volume sera utilisé plus tard dans la fonction de coût du processus d'optimisation associé au problème d'annexion superellipsoïdale.

**Proposition 2.7** (volume d'un superellipsoïde  $n$ -D déformé). *Soit  $\mathcal{S}$  un superellipsoïde déformé de dimension  $n$ , décrit par le vecteur  $\Psi$  défini par (2.23). Le volume  $V_n(\Psi)$  de  $\mathcal{S}$*



est donné par

$$V_n(\Psi) = 2a_n \left[ \prod_{\substack{i=1 \\ i \neq p-1}}^{n-1} a_i \varepsilon_i B\left(\frac{\varepsilon_i}{2}, i \frac{\varepsilon_i}{2} + 1\right) \right] \cdot \left[ a_{p-1} \varepsilon_{p-1} \sum_{|\alpha|=0}^{n-1} v^\alpha B\left(\frac{|\alpha|+1}{2} \varepsilon_{p-1}, \frac{p-1}{2} \varepsilon_{p-1} + 1\right) \right], \quad (2.30)$$

où le multi-entier  $\alpha = (\alpha_1, \dots, \alpha_{p-1}, 0, \alpha_{p+1}, \dots, \alpha_n)$  vérifie

$$v^\alpha = \prod_{k=1}^n v_k^{\alpha_k}, \quad |\alpha| = \sum_{j=1}^n \alpha_j, \quad \alpha_i \in \{0, 1\}, \quad i = 1, \dots, n,$$

De plus, la fonction Beta  $B(x, y)$  est liée à la fonction Gamma par

$$B(x, y) = 2 \int_0^{\pi/2} \sin^{2x-1} \phi \cos^{2y-1} \phi d\phi = \frac{\Gamma(x)\Gamma(y)}{\Gamma(x+y)},$$

la fonction Gamma étant typiquement définie par

$$\Gamma(x) = \int_0^\infty \exp^{-t} t^{x-1} dt,$$

où, de manière équivalente, par la somme discrète

$$\Gamma(x) = \lim_{n \rightarrow +\infty} \frac{n! n^x}{(x+1) \dots (x+n)}.$$

**Preuve.** Nous allons commencer par démontrer l'expression (2.30) en dimension 3, puis nous étendrons ensuite la démarche à la dimension  $n$ .

### Démonstration en dimension 3

Considérons une superellipse, dont la représentation trigonométrique explicite est donnée par

$$\begin{cases} x = a \cos^{\varepsilon_1} \theta_1 \\ y = b \sin^{\varepsilon_1} \theta_1 \end{cases}, \quad -\pi \leq \theta_1 < \pi. \quad (2.31)$$

L'aire de la superellipse décrite par (2.31) se calcule par la formule de Green [149]

$$A = \oint_{\mathcal{C}'} (x dy - y dx), \quad (2.32)$$

le long du contour  $\mathcal{C}'$  représenté sur la figure 2.11. Du fait de la propriété de symétrie de la superellipse dans le plan  $(Oxy)$ , on peut simplifier l'intégrale (2.32) en considérant uniquement le contour fermé  $\mathcal{C}$ , divisé en trois segments  $C_1$ ,  $C_2$  et  $C_3$  (voir figure 2.11). On obtient alors

$$\begin{aligned} A &= 2 \oint_{\mathcal{C}} (x dy - y dx) \\ &= 2 \int_{C_1} (x dy - y dx) + 2 \int_{C_2} (x dy - y dx) + 2 \int_{C_3} (x dy - y dx). \end{aligned}$$

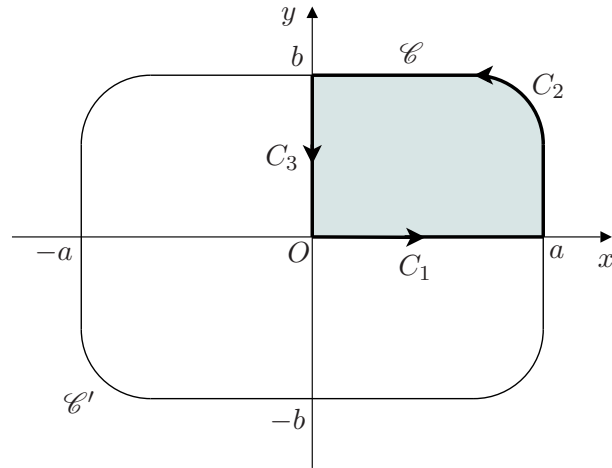


FIG. 2.11 – contours d'intégration considérés pour déterminer l'aire une superellipse.

L'intégrale le long de  $C_1$  est nulle car  $y = 0$  et  $dy = 0$ . De même, l'intégrale le long de  $C_3$  est nulle. On obtient donc :

$$\begin{aligned}
 A &= 2 \int_{C_2} (x dy - y dx) \\
 &= 2 \int_0^{\pi/2} (x \dot{y} - y \dot{x}) d\theta_1 \\
 &= 2 \int_0^{\pi/2} (\sin^{\varepsilon_1-1} \theta_1 \cos^{\varepsilon_1+1} \theta_1 + \sin^{\varepsilon_1+1} \theta_1 \cos^{\varepsilon_1-1} \theta_1) d\theta_1 \\
 &= ab\varepsilon_1 \left[ B\left(\frac{\varepsilon_1}{2}, \frac{\varepsilon_1+2}{2}\right) + B\left(\frac{\varepsilon_1+2}{2}, \frac{\varepsilon_1}{2}\right) \right] \\
 &= 2ab\varepsilon_1 B\left(\frac{\varepsilon_1}{2}, \frac{\varepsilon_1+2}{2}\right), \tag{2.33}
 \end{aligned}$$

où la fonction Beta  $B(x, y)$  est définie par

$$B(x, y) = 2 \int_0^{\pi/2} \sin^{2x-1} \phi \cos^{2y-1} \phi d\phi = \frac{\Gamma(x)\Gamma(y)}{\Gamma(x+y)},$$

et la fonction Gamma par

$$\Gamma(x) = \int_0^{\infty} \exp^{-t} t^{x-1} dt,$$

où encore, par la somme discrète

$$\Gamma(x) = \lim_{n \rightarrow +\infty} \frac{n! n^x}{(x+1) \dots (x+n)}.$$

Considérons à présent un superellipsoïde de dimension 3, déformé sous l'action d'un pincement linéaire le long de l'axe  $z$ , dont la paramétrisation trigonométrique explicite (cf

définition 2.4) est donnée par

$$\begin{cases} x = a_1(v_1 \sin^{\varepsilon_2} \theta_2 + 1) \cos^{\varepsilon_1} \theta_1 \cos^{\varepsilon_2} \theta_2, \\ y = a_2(v_2 \sin^{\varepsilon_2} \theta_2 + 1) \sin^{\varepsilon_1} \theta_1 \cos^{\varepsilon_2} \theta_2, \\ z = a_3 \sin^{\varepsilon_2} \theta_2. \end{cases}$$

Ce superellipsoïde 3D peut être vu comme un empilement de superellipses d'épaisseur infinitésimale  $dz$  (voir figure 2.12). D'après l'équation (2.33), l'aire de la section du superellipsoïde située à l'élévation  $z$  est alors donnée par

$$A(\theta_2) = 2a(\theta_2)b(\theta_2)\varepsilon_1 B\left(\frac{\varepsilon_1}{2}, \frac{\varepsilon_1 + 2}{2}\right),$$

avec

$$\begin{aligned} a(\theta_2) &= a_1 \cos^{\varepsilon_2} \theta_2 (1 + v_1 \sin^{\varepsilon_2} \theta_2), \\ b(\theta_2) &= a_2 \cos^{\varepsilon_2} \theta_2 (1 + v_2 \sin^{\varepsilon_2} \theta_2). \end{aligned}$$

Le volume infinitésimal  $dV$  est donné par

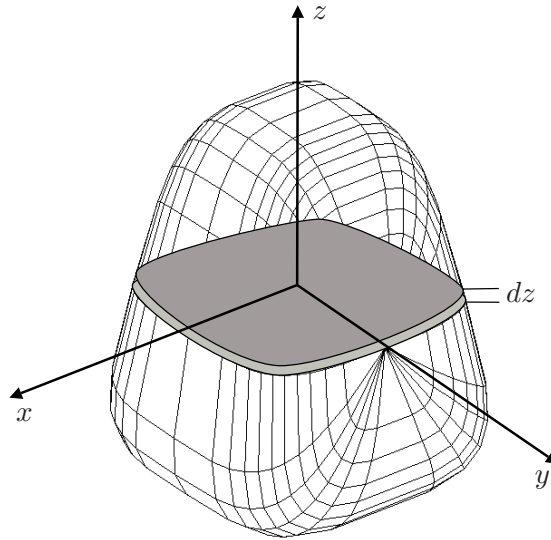


FIG. 2.12 – interprétation d'un superellipsoïde 3D comme un empilement de superellipses d'épaisseurs infinitésimales  $dz$ .

$$dV = A(z)dz = A(\theta_2)\dot{z}(\theta_2)d\theta_2,$$

avec

$$\dot{z}(\theta_2) = (a_3 \sin^{\varepsilon_2} \theta_2)' = a_3 \varepsilon_2 \sin^{\varepsilon_2-1} \theta_2 \cos \theta_2,$$

et

$$A(\theta_2) = 2a_1 a_2 \varepsilon_1 \cos^{2\varepsilon_2} \theta_2 (1 + v_1 \sin^{\varepsilon_2} \theta_2) (1 + v_2 \sin^{\varepsilon_2} \theta_2) B\left(\frac{\varepsilon_1}{2}, \frac{\varepsilon_1 + 2}{2}\right).$$

On obtient donc

$$dV = 2a_1 a_2 a_3 \varepsilon_1 \varepsilon_2 B\left(\frac{\varepsilon_1}{2}, \frac{\varepsilon_1 + 2}{2}\right) \sin^{\varepsilon_2-1} \theta_2 \cos^{2\varepsilon_2+1} \theta_2 (1 + v_1 \sin^{\varepsilon_2} \theta_2) (1 + v_2 \sin^{\varepsilon_2} \theta_2) d\theta_2.$$

Le volume  $V$  est alors obtenu en intégrant  $dV$  le long de l'anomalie  $\theta_2 \in [-\pi/2, \pi/2]$ , tel que

$$V = \int_{-\pi/2}^{\pi/2} A(\theta_2) \dot{z}(\theta_2) d\theta_2,$$

soit

$$V = 2a_1 a_2 a_3 \varepsilon_1 \varepsilon_2 B\left(\frac{\varepsilon_1}{2}, \frac{\varepsilon_1 + 2}{2}\right) \int_{-\pi/2}^{\pi/2} \sin^{\varepsilon_2-1} \theta_2 \cos^{2\varepsilon_2+1} \theta_2 (1 + v_1 \sin^{\varepsilon_2} \theta_2) (1 + v_2 \sin^{\varepsilon_2} \theta_2) d\theta_2.$$

En remarquant que

$$\begin{aligned} & \int_{-\pi/2}^{\pi/2} \sin^{\varepsilon_2-1} \theta_2 \cos^{2\varepsilon_2+1} \theta_2 (1 + v_1 \sin^{\varepsilon_2} \theta_2) (1 + v_2 \sin^{\varepsilon_2} \theta_2) d\theta_2 = B\left(\frac{\varepsilon_2}{2}, \varepsilon_2 + 1\right) \\ & + (v_1 + v_2) B(\varepsilon_2, \varepsilon_2 + 1) + v_1 v_2 B\left(\frac{3\varepsilon_2}{2}, \varepsilon_2 + 1\right), \end{aligned}$$

il vient :

$$\begin{aligned} V = & 2a_1 a_2 a_3 \varepsilon_1 \varepsilon_2 B\left(\frac{\varepsilon_1}{2}, \frac{\varepsilon_1 + 2}{2}\right) \left[ B\left(\frac{\varepsilon_2}{2}, \varepsilon_2 + 1\right) + (v_1 + v_2) B(\varepsilon_2, \varepsilon_2 + 1) \right. \\ & \left. + v_1 v_2 B\left(\frac{3\varepsilon_2}{2}, \varepsilon_2 + 1\right) \right]. \end{aligned} \quad (2.34)$$

Finalement, en introduisant la notation par multi-entiers, l'expression (2.34) peut se ré-écrire de manière plus compacte sous la forme

$$V = 2a_3 a_1 \varepsilon_1 B\left(\frac{\varepsilon_1}{2}, \frac{\varepsilon_1 + 2}{2}\right) a_2 \varepsilon_2 \sum_{|\alpha|=0}^2 v^\alpha B\left(\frac{|\alpha| + 1}{2} \varepsilon_2, \varepsilon_2 + 1\right).$$

où le multi-entier  $\alpha = (\alpha_1, \alpha_2, 0)$  vérifie

$$v^\alpha = \prod_{k=1}^3 v_k^{\alpha_k}, \quad |\alpha| = \sum_{j=1}^3 \alpha_j, \quad \alpha_i \in \{0, 1\}, \quad i = 1, \dots, 3.$$

### Démonstration en dimension $n$

Supposons que le volume  $V_n$  d'un superellipsoïde *non déformé* de dimension  $n$ , s'écrit

$$V_n = 2a_n \prod_{i=1}^{n-1} a_i \varepsilon_i B\left(\frac{\varepsilon_i}{2}, i\frac{\varepsilon_i}{2} + 1\right),$$

et cherchons à exprimer le volume  $V_{n+1}$  d'un superellipsoïde *déformé* de dimension  $n + 1$ .

Comme pour le calcul du volume en trois dimensions, nous considérons que les demi-grand axes de  $V_n$  dépendent de la nouvelle anomalie  $\theta_n$  associée au volume  $V_{n+1}$ , et des coefficients de pincement linéaire  $v = (v_1, v_2, \dots, v_n)$ . Soit :

$$V_n(\theta_n) = 2a'_n \prod_{i=1}^{n-1} a'_i \varepsilon_i B\left(\frac{\varepsilon_i}{2}, i\frac{\varepsilon_i}{2} + 1\right),$$

avec

$$a'_i = a_i \cos^{\varepsilon_n} \theta_n (1 + v_i \sin^{\varepsilon_n} \theta_n), \quad i = 1, \dots, n - 1.$$

Le volume infinitésimal  $dV_{n+1}$  s'écrit sous la forme

$$dV_{n+1} = V_n(\theta_n) \dot{x}_{n+1}(\theta_n) d\theta_n, \quad (2.35)$$

où  $x_{n+1}$  s'écrit

$$x_{n+1}(\theta_n) = a_{n+1} \sin^{\varepsilon_n} \theta_n, \quad (2.36)$$

d'après la paramétrisation trigonométrique explicite (2.24).

**Remarque 2.8.** Afin de simplifier la démonstration, nous considérons que le pincement linéaire est appliqué sur l'axe  $x_{n+1}$ , mais la démarche reste valable pour une direction arbitraire. Notamment, nous donnerons l'expression finale du volume du superellipsoïde pour une direction de pincement quelconque.

La dérivée de l'expression (2.36) s'écrit

$$\dot{x}_{n+1}(\theta_n) = \varepsilon_n a_{n+1} \sin^{\varepsilon_n - 1} \theta_n \cos \theta_n,$$

et (2.35) peut donc se réécrire

$$\begin{aligned} dV_{n+1} &= 2a_{n+1} a_n \varepsilon_n \sin^{\varepsilon_n - 1} \theta_n \cos^{\varepsilon_n + 1} \theta_n (1 + v_n \sin^{\varepsilon_n} \theta_n) \\ &\times \prod_{i=1}^{n-1} a_i \varepsilon_i \cos^{\varepsilon_n} \theta_n (1 + v_i \sin^{\varepsilon_n} \theta_n) B\left(\frac{\varepsilon_i}{2}, i\frac{\varepsilon_i}{2} + 1\right). \end{aligned} \quad (2.37)$$

Le volume  $V_{n+1}$  s'obtient alors en intégrant (2.37) le long de la nouvelle anomalie  $\theta_n \in [-\pi/2, \pi/2]$ , tel que

$$V_{n+1} = \int_{-\pi/2}^{\pi/2} V_n(\theta_n) \dot{x}_{n+1}(\theta_n) d\theta_n.$$

On obtient alors

$$\begin{aligned}
 V_{n+1} &= 2a_{n+1} \left[ \prod_{i=1}^{n-1} a_i \varepsilon_i B \left( \frac{\varepsilon_i}{2}, i \frac{\varepsilon_i}{2} + 1 \right) \right] \int_{-\pi/2}^{\pi/2} \left[ a_n \varepsilon_n \cos^{\varepsilon_n+1} \theta_n \sin^{\varepsilon_n-1} \theta_n \prod_{i=1}^{n-1} \cos^{\varepsilon_i} \theta_n \right. \\
 &\quad \left. \times \prod_{i=1}^n (1 + v_i \sin^{\varepsilon_i} \theta_n) \right] d\theta_n \\
 V_{n+1} &= 2a_{n+1} \left[ \prod_{i=1}^{n-1} a_i \varepsilon_i B \left( \frac{\varepsilon_i}{2}, i \frac{\varepsilon_i}{2} + 1 \right) \right] \cdot \left[ a_n \varepsilon_n \int_{-\pi/2}^{\pi/2} \cos^{n\varepsilon_n+1} \theta_n \sin^{\varepsilon_n-1} \theta_n \right. \\
 &\quad \left. \times \prod_{i=1}^n (1 + v_i \sin^{\varepsilon_i} \theta_n) d\theta_n \right] \tag{2.38}
 \end{aligned}$$

En utilisant la notation par multi-entiers, l'expression (2.38) peut alors se réécrire

$$V_{n+1} = 2a_{n+1} \left[ \prod_{i=1}^{n-1} a_i \varepsilon_i B \left( \frac{\varepsilon_i}{2}, i \frac{\varepsilon_i}{2} + 1 \right) \right] \cdot \left[ a_n \varepsilon_n \sum_{|\alpha|=0}^n v^\alpha \int_{-\pi/2}^{\pi/2} \sin^{(|\alpha|+1)\varepsilon_n-1} \theta_n \cos^{n\varepsilon_n+1} \theta_n d\theta_n \right],$$

où le multi-entier  $\alpha = (\alpha_1, \alpha_2, \dots, \alpha_n, 0)$  vérifie

$$v^\alpha = \prod_{k=1}^{n+1} v_k^{\alpha_k}, \quad |\alpha| = \sum_{j=1}^{n+1} \alpha_j, \quad \alpha_i \in \{0, 1\}, \quad i = 1, \dots, n+1.$$

Or, par définition de la fonction Beta, on a

$$\int_{-\pi/2}^{\pi/2} \sin^{(|\alpha|+1)\varepsilon_n-1} \theta_n \cos^{n\varepsilon_n+1} \theta_n d\theta_n = B \left( \frac{|\alpha|+1}{2} \varepsilon_n, \frac{n}{2} \varepsilon_n + 1 \right).$$

D'où :

$$V_{n+1} = 2a_{n+1} \left[ \prod_{i=1}^{n-1} a_i \varepsilon_i B \left( \frac{\varepsilon_i}{2}, i \frac{\varepsilon_i}{2} + 1 \right) \right] \cdot \left[ a_n \varepsilon_n \sum_{|\alpha|=0}^n v^\alpha B \left( \frac{|\alpha|+1}{2} \varepsilon_n, \frac{n}{2} \varepsilon_n + 1 \right) \right]. \tag{2.39}$$

En remarquant que la direction de pincement choisie est  $p = n + 1$ , on peut réécrire l'expression (2.39) sous la forme

$$V_{n+1} = 2a_{n+1} \left[ \prod_{i=1}^{n-1} a_i \varepsilon_i B \left( \frac{\varepsilon_i}{2}, i \frac{\varepsilon_i}{2} + 1 \right) \right] \cdot \left[ a_{p-1} \varepsilon_{p-1} \sum_{|\alpha|=0}^n v^\alpha B \left( \frac{|\alpha|+1}{2} \varepsilon_{p-1}, \frac{p-1}{2} \varepsilon_{p-1} + 1 \right) \right]. \tag{2.40}$$

Finalement, en choisissant une direction de pincement  $p$  arbitraire,  $2 \leq p \leq n + 1$ , le volume (2.40) s'écrit

$$V_{n+1} = 2a_{n+1} \left[ \prod_{\substack{i=1 \\ i \neq p-1}}^n a_i \varepsilon_i B \left( \frac{\varepsilon_i}{2}, i \frac{\varepsilon_i}{2} + 1 \right) \right] \cdot \left[ a_{p-1} \varepsilon_{p-1} \sum_{|\alpha|=0}^n v^\alpha B \left( \frac{|\alpha|+1}{2} \varepsilon_{p-1}, \frac{p-1}{2} \varepsilon_{p-1} + 1 \right) \right], \tag{2.41}$$

où le multi-entier  $\alpha = (\alpha_1, \dots, \alpha_{p-1}, 0, \alpha_{p+1}, \dots, \alpha_{n+1})$  vérifie

$$v^\alpha = \prod_{k=1}^{n+1} v_k^{\alpha_k}, \quad |\alpha| = \sum_{j=1}^{n+1} \alpha_j, \quad \alpha_i \in \{0, 1\}, \quad i = 1, \dots, n+1.$$

La relation de récurrence (2.41) étant vraie pour  $n+1$ , elle est donc vraie pour tout  $n$ , ce qui termine la preuve.  $\blacksquare$

De plus, afin d'éviter des problèmes numériques lors de la résolution du problème d'optimisation, un *volume normalisé*  $\tilde{V}_n$  est utilisé de manière préférentielle, tel que

$$\tilde{V}_n(\Psi) = V_n(\Psi)^{\frac{1}{n}}, \quad (2.42)$$

où  $n$  est l'ordre de la forme superquadrique.

**Corollaire 2.1.** *Le volume d'un superellipsoïde non déformé de dimension  $n$ , défini par le vecteur de demi-grand axes  $a \in \mathbb{R}^n$  et par le vecteur de paramètres de forme  $\varepsilon \in \mathbb{R}^{n-1}$ , est donné par*

$$V_n = 2a_n \prod_{i=1}^{n-1} a_i \varepsilon_i B\left(\frac{\varepsilon_i}{2}, i\frac{\varepsilon_i}{2} + 1\right).$$

Finalement, la *distance radiale euclidienne* en dimension  $n$  peut être utile afin de déterminer la plus courte distance entre un point arbitraire et sa projection sur la surface du superellipsoïde. Plus précisément, cette distance sera utilisée dans le problème d'optimisation de trajectoires de rentrée atmosphérique, afin de vérifier que les conditions de bords du problème de commande optimale sont bien situées à l'intérieur (ou à la surface) de la forme convexe.

**Proposition 2.8** (distance radiale euclidienne  $n$ -D). *La distance radiale euclidienne  $d(\Psi, x_0)$  est définie comme étant la distance entre un point  $Q$  de coordonnées  $x_0$ , et un point  $P$  de coordonnées  $x_s$ , correspondant à la projection de  $Q$  sur la surface du superellipsoïde le long de la direction formée par le point  $Q$  et le centre de la forme géométrique. Pour un superellipsoïde quelconque, décrit par le vecteur  $\Psi$  défini par (2.23), l'expression de la distance radiale euclidienne  $d(\Psi, x_0) = |x_0 - x_s|$  est donnée par*

$$d(\Psi, x_0) = |x_0| \cdot \left| 1 - (\mathcal{F}_n(\Psi, x_0))^{-\frac{\varepsilon_{n-1}}{2}} \right|, \quad (2.43)$$

où la fonction d'appartenance  $\mathcal{F}_n(\Psi, x_0)$  est définie par (2.26).

**Preuve.** Soit  $x_0 = (x_{10}, x_{20}, \dots, x_{n0})$  les coordonnées d'un point  $Q$ , situé à l'intérieur ou à l'extérieur du superellipsoïde, et soit

$$\mathcal{F}_n(\Psi, x_0) = \Lambda_{n,n}(\Psi, x_0)$$

la fonction d'appartenance du point  $Q$  au superellipsoïde déformé, dont l'expression générale est donnée par (2.26).

Soit  $x_s = \beta x_0$  un point  $P$  situé à la surface du superellipsoïde.

D'après (2.26), pour l'itération  $k = 2$ , on a

$$\begin{aligned}\Lambda_{n,2}(\Psi, \beta x_0) &= \left( \frac{\beta x_{10}}{a_1 \left( \frac{v_1}{a_p} x_{1p} + 1 \right)} \right)^{\frac{2}{\varepsilon_1}} + \left( \frac{\beta x_{20}}{a_2 \left( \frac{v_2}{a_p} x_{1p} + 1 \right)} \right)^{\frac{2}{\varepsilon_1}} \\ &= \beta^{\frac{2}{\varepsilon_1}} \left[ \left( \frac{x_{10}}{a_1 \left( \frac{v_1}{a_p} x_{1p} + 1 \right)} \right)^{\frac{2}{\varepsilon_1}} + \left( \frac{x_{20}}{a_2 \left( \frac{v_2}{a_p} x_{1p} + 1 \right)} \right)^{\frac{2}{\varepsilon_1}} \right] \\ &= \beta^{\frac{2}{\varepsilon_1}} \Lambda_{n,2}(\Psi, x_0).\end{aligned}$$

Ensuite, pour l'itération  $k$ , supposons que  $\Lambda_{n,k-1}(\Psi, \beta x_0)$  s'écrive

$$\Lambda_{n,k-1}(\Psi, \beta x_0) = \beta^{\frac{2}{\varepsilon_{k-2}}} \Lambda_{n,k-1}(\Psi, x_0), \quad (2.44)$$

et cherchons à déterminer l'expression de  $\Lambda_{n,k}(\Psi, \beta x_0)$ . Alors, d'après (2.26), on a

$$\begin{aligned}\Lambda_{n,k}(\Psi, \beta x_0) &= (\Lambda_{n,k-1}(\Psi, \beta x_0))^{\frac{\varepsilon_{k-2}}{\varepsilon_{k-1}}} + \left( \frac{\beta x_{k0}}{a_k \left( \frac{v_k}{a_p} x_{1p} + 1 \right)} \right)^{\frac{2}{\varepsilon_{k-1}}} \\ &= \left( \beta^{\frac{2}{\varepsilon_{k-2}}} \Lambda_{n,k-1}(\Psi, x_0) \right)^{\frac{\varepsilon_{k-2}}{\varepsilon_{k-1}}} + \beta^{\frac{2}{\varepsilon_{k-1}}} \left( \frac{x_{k0}}{a_k \left( \frac{v_k}{a_p} x_{1p} + 1 \right)} \right)^{\frac{2}{\varepsilon_{k-1}}} \\ &= \beta^{\frac{2}{\varepsilon_{k-1}}} \left[ (\Lambda_{n,k-1}(\Psi, x_0))^{\frac{\varepsilon_{k-2}}{\varepsilon_{k-1}}} + \left( \frac{x_{k0}}{a_k \left( \frac{v_k}{a_p} x_{1p} + 1 \right)} \right)^{\frac{2}{\varepsilon_{k-1}}} \right] \\ &= \beta^{\frac{2}{\varepsilon_{k-1}}} \Lambda_{n,k}(\Psi, x_0).\end{aligned}$$

La relation de récurrence (2.44) étant vraie pour l'itération  $k$ , elle est donc vraie pour tout  $k$ . En particulier, pour  $k = n$ , on obtient

$$\begin{aligned}\mathcal{F}_n(\Psi, \beta x_0) &= \Lambda_{n,n}(\Psi, \beta x_0) \\ &= \beta^{\frac{2}{\varepsilon_{n-1}}} \Lambda_{n,n}(\Psi, x_0).\end{aligned}$$

Or, par définition de la fonction d'appartenance,

$$\mathcal{F}_n(\Psi, \beta x_0) = 1,$$

d'où

$$\begin{aligned}\mathcal{F}_n(\Psi, x_0) &= \Lambda_{n,n}(\Psi, x_0) \\ &= \beta^{-\frac{2}{\varepsilon_{n-1}}}.\end{aligned}$$



Finalement, la distance radiale euclidienne s'écrit

$$\begin{aligned}
 d(\Psi, x_0) &= |x_0 - x_s| \\
 &= |x_0 - \beta x_0| \\
 &= |x_0| \cdot |1 - \beta| \\
 &= |x_0| \cdot \left| 1 - (\mathcal{F}_n(\Psi, x_0))^{-\frac{\varepsilon_{n-1}}{2}} \right|,
 \end{aligned}$$

ce qui termine la preuve. ■

**Corollaire 2.2.** *La distance radiale euclidienne  $d(x_0)$  entre un point  $Q$  de coordonnées  $x_0$  et sa projection  $P$  à la surface d'un superellipsoïde non déformé de dimension  $n$ , le long d'une droite passant par le point  $Q$  et le centre de la superquadrique, est définie par*

$$d(x_0) = |x_0| \cdot \left| 1 - (\mathcal{F}_n(x_0))^{-\frac{\varepsilon_{n-1}}{2}} \right|,$$

où la fonction d'appartenance  $\mathcal{F}_n(x_0)$  est définie par (2.9).

## 2.4.2 Problème d'annexion superellipsoïdale

Nous sommes maintenant en mesure de poser le problème d'approximation intérieure convexe d'un domaine atteignable non convexe par un ellipsoïde superquadrique. Ce n'est pas un problème d'approximation classique dans le sens où le domaine non convexe que l'on cherche à approximer est décrit par un ensemble d'équations implicites, i.e., il n'admet pas de représentation explicite dans le cas général.

Sans perte de généralité, nous supposons que le domaine non convexe peut être décrit au moyen d'une ou plusieurs relations analytiques implicites du type

$$f_{min} \leq f_{nc}(x) \leq f_{max}.$$

où  $x$  est un ensemble de variables de dimension  $n$ . Le problème d'annexion superellipsoïdale est alors défini comme suit.

**Problème 2.1** (problème d'annexion superellipsoïdale). *Soit  $\mathcal{S}$  un superellipsoïde de dimension  $n$ , décrit par le vecteur  $\Psi$  défini par (2.23). Le problème d'annexion superellipsoïdale consiste alors à déterminer les paramètres optimaux  $\Psi^*$  associés au superellipsoïde de plus grand volume  $\mathcal{S}_{opt}$  contenu dans le domaine atteignable (supposé être non convexe) décrit par la relation analytique  $f_{nc}$ , tel que*

$$\begin{aligned}
 \max_{\Psi} \quad & \tilde{V}_n(\Psi) & (2.45) \\
 t.q. \quad & \begin{cases} \mathcal{F}_n(\Psi, x) \leq 1, \\ f_{min} \leq f_{nc}(x) \leq f_{max}, \\ x_i^l \leq x_i \leq x_i^u, \quad i = 1, \dots, n. \end{cases}
 \end{aligned}$$

où le volume du superellipsoïde  $\tilde{V}_n(\Psi)$  est défini par (2.42), et la fonction d'appartenance  $\mathcal{F}_n(\Psi, x)$  par (2.26). Les variables  $x$  représentent les coordonnées cartésiennes associées à un nombre donné de points<sup>41</sup> quadrillant la surface du superellipsoïde.

Dans la section suivante, nous détaillons le processus d'optimisation utilisé afin de déterminer l'ensemble des paramètres de dimensionnement  $\Psi^*$  de la superquadrique déformée. A cet effet, nous avons choisi de considérer une classe d'algorithmes évolutionnistes, appelés algorithmes génétiques, bien connu pour leur propriété de convergence stochastique vers une solution globale.

### 2.4.3 Résolution par algorithme génétique

#### Rappels sur les algorithmes génétiques

Depuis quelques années, un intérêt grandissant de la part de la communauté scientifique s'est manifesté afin d'appliquer les algorithmes génétiques (GA) aux problèmes d'optimisation issus du monde industriel. Comparé aux techniques traditionnelles de recherche et d'optimisation, telles que les méthodes à base de gradients et les stratégies énumératives, les algorithmes génétiques sont robustes et convergent le plus souvent vers une solution globale de manière asymptotique. De plus, ils sont très facilement applicables aux situations où très peu de connaissances sont disponibles concernant le problème à optimiser. Enfin, pour certains types d'algorithmes génétiques classiques, des conditions nécessaires et suffisantes d'optimalité et de convergence asymptotique globale ont été obtenues dans [150, 151]. Les algorithmes génétiques s'inspirent du processus régissant l'évolution naturelle des organismes biologiques : sur plusieurs générations, les populations naturelles évoluent selon les principes de la sélection naturelle et de la "survie du plus fort" [152]. Contrairement aux méthodes d'optimisation fondées sur l'utilisation de gradients, les GA n'utilisent pas de calculs de Jacobiens, ou même un estimé initial formel de l'espace solution. En outre, de part la nature stochastique du mécanisme de recherche, les algorithmes génétiques sont capables de parcourir l'ensemble de l'espace solution, avec plus de chances d'approcher l'optimum global. Finalement, même si les algorithmes génétiques ne sont pas capables de fournir une valeur exacte de la solution optimale, la meilleure solution obtenue peut être affinée en utilisant une méthode d'optimisation locale (algorithme de type Newton par exemple).

Typiquement, un algorithme génétique peut être caractérisé par les cinq éléments suivants :

1. un *mécanisme d'encodage* de chaque individu dans la population<sup>42</sup> de type binaire, à valeurs réelles, etc.
2. un processus permettant de générer la *population initiale*, soit de manière arbitraire, ou basé sur une connaissance *a priori* du problème.
3. une fonction à optimiser, appelée *aptitude*, où fonction d'évaluation des individus,

---

<sup>41</sup>ce nombre est fixé en fonction du partitionnement du vecteur d'anomalies  $\theta$ .

<sup>42</sup>rappelons, dans le formalisme des algorithmes génétiques, qu'une population est constituée d'individus, eux-mêmes formés de gènes, qui correspondent alors aux variables d'optimisation du problème.

4. des *opérateurs génétiques* : principalement croisement, mutation, migration, sélection.
5. des *paramètres de dimensionnement* : nombre d'individus dans la population, nombre de générations, taux d'application des opérateurs génétiques, etc.

La figure 2.13 illustre la structure classique d'un algorithme génétique. Au début du processus d'optimisation, la population initiale est créée. Un certain nombre d'individus sont alors sélectionnés dans l'espace de recherche, puis leurs aptitudes évaluées. Ensuite, tant que le critère d'arrêt<sup>43</sup> de l'algorithme n'est pas satisfait, de nouvelles générations de descendants sont produites. A cette fin, des individus sont sélectionnés dans la population précédente, par exemple en fonction de leur aptitude, afin de produire des descendants. Cette dernière opération est réalisée grâce à des opérateurs génétiques de croisement et de mutation. Ensuite, l'aptitude de ces descendants est évaluée, et ils sont réinsérés dans la population, souvent remplaçant les parents possédant les aptitudes les plus faibles.

Les algorithmes génétiques basés sur des populations uniques sont souvent puissant, et se comporte de manière satisfaisante pour la plupart des problèmes d'optimisation rencontrés. Dans le cas des algorithmes génétiques à populations multiples (où multi-populations), chaque sous-population évolue séparément sur une échelle de temps donnée, c'est à dire sur un nombre de générations donné, avant qu'un ou plusieurs individus ne soient échangés entre les différentes sous-populations selon une topologie de *migration* prédéfinie. Les GA multi-populations permettent notamment de balayer de manière plus efficace l'espace de recherche afin de converger plus rapidement vers la solution globale. Le lecteur est renvoyé, par exemple, aux ouvrages [153, 154] pour obtenir plus de précisions concernant les algorithmes génétiques.

### Résolution du problème d'annexion superellipsoïdale

Dans le cadre de la résolution du problème d'annexion superellipsoïdale 2.1, l'idée retenue est alors d'utiliser un algorithme génétique hors ligne afin de construire une approximation intérieure convexe d'un domaine atteignable non convexe. Dans ce cas, la population initiale de l'algorithme génétique est construite de façon à ce que des formes convexes élémentaires satisfaisant les contraintes (2.45) soient générées de manière aléatoire sur tout l'espace de recherche (voir figure 2.14). Tant que le critère d'arrêt n'est pas satisfait, une nouvelle génération est créée. Un certain nombre d'individus de chaque sous-population est alors choisi grâce à une méthode de sélection par tournois, puis ces individus sont croisés et mutés en utilisant des opérateurs génétiques adaptés. Les descendants sont alors évalués, puis réinsérés dans l'ensemble de la population. Des stratégies de migration peuvent alors être mises en place entre les différentes sous-populations.

**Remarque 2.9.** Généralement, le choix de la population initiale affecte fortement la convergence des algorithmes génétiques. Si l'emplacement du minimum global à l'intérieur de l'espace de recherche est complètement inconnu, il semble alors être naturel de générer des individus aléatoirement pour couvrir l'ensemble du domaine de recherche.

---

<sup>43</sup>par exemple, le critère d'arrêt peut être choisi comme étant un nombre prédéfini de générations à atteindre, où une stagnation de l'aptitude moyenne de la population sur un nombre fini de générations.

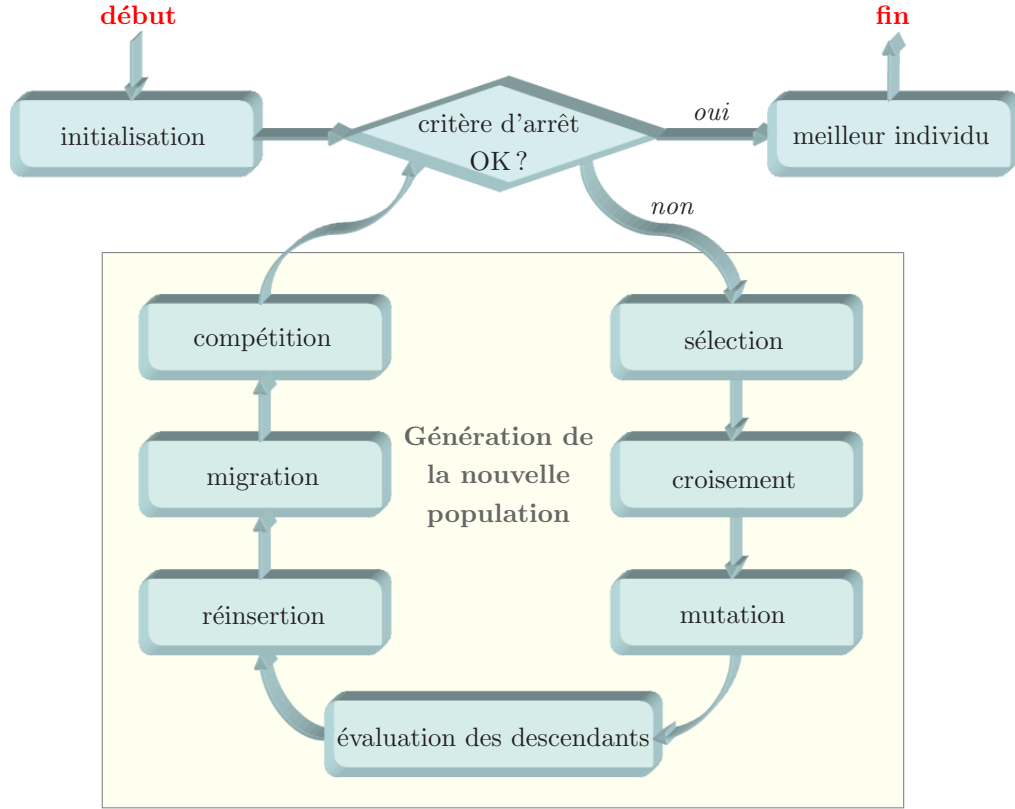


FIG. 2.13 – organigramme d'un algorithme génétique classique.

Au contraire, si une connaissance *a priori* du minimiseur est disponible, il semble alors évident de générer la population initiale uniquement dans un sous-espace du domaine de recherche de manière à accélérer la convergence du processus.

La méthode de satisfaction de contraintes retenue pour l'algorithme génétique n'est pas fondée sur des fonctions de pénalité classiques, couramment utilisées dans les algorithmes d'optimisation ordinaires. En fait, le principal inconvénient associé à l'utilisation de paramètres de pénalité est que leur choix dépend fortement de l'application considérée [155]. Nous avons alors choisi d'utiliser un opérateur de *sélection par tournois* [156], où chaque individu est évalué par son aptitude  $F_i$ ,  $i = 1, \dots, n_{ind}$ , définie par

$$F_i(x) = \begin{cases} f_i(x) & \text{si } \Phi_j(x) \leq 0, \quad \forall j = 1, \dots, n_c, \\ f_{worst} + \sum_{j=1}^{n_c} \langle \Phi_j(x) \rangle, & \text{sinon,} \end{cases} \quad (2.46)$$

où  $n_c$  représente le nombre de contraintes normalisées  $\Phi_j$ ,  $n_{ind}$  est le nombre d'individus, et où l'aptitude  $f_i$  est égale à l'inverse du volume du superellipsoïde (2.42), i.e.,

$$f_i = \frac{1}{\tilde{V}_n(\Psi_i)}, \quad i = 1, \dots, n_{ind}.$$

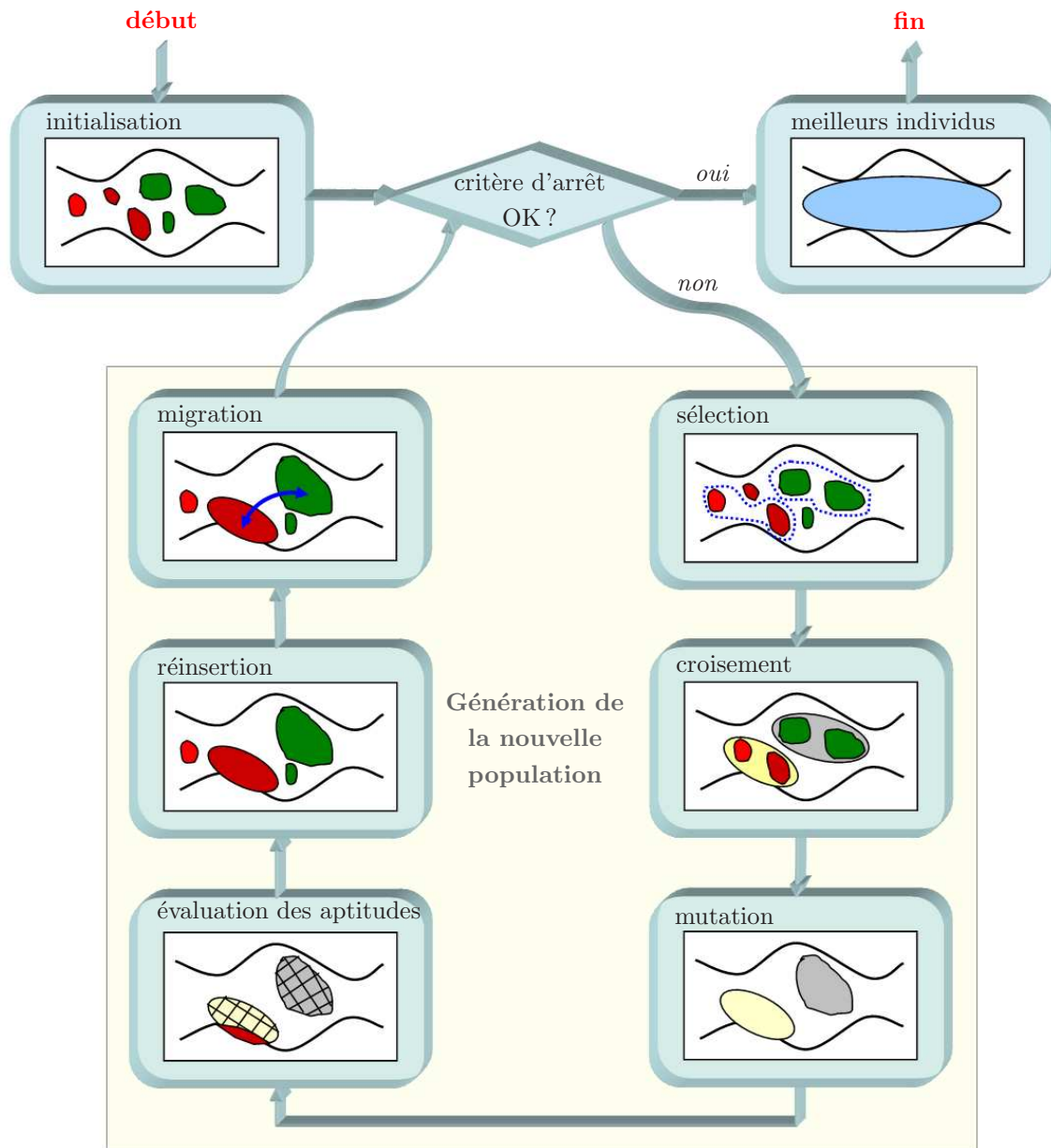


FIG. 2.14 – résolution du problème d’annexion superellipsoïdale par algorithme génétique.

Par exemple, si le domaine non convexe que l’on cherche à approximer est défini par une relation implicite du type  $f_{min} \leq f_{nc}(x) \leq f_{max}$ , on peut alors choisir  $\Phi_1(x) = f_{min} - f_{nc}(x)$  et  $\Phi_2(x) = f_{max} - f_{nc}(x)$ . En outre, la valeur  $f_{worst}$ , correspondant à la pire solution admissible dans la population, est définie par

$$f_{worst} = \max_i \left( \frac{1}{\tilde{V}_n(\Psi_i)} \right), \quad i = 1, \dots, n_{ind}.$$

Ensuite, deux solutions, c’est-à-dire deux individus, sont comparés entre eux et les critères

suivant sont alors systématiquement appliqués.

- toute solution faisable est préférée à toute solution infaisable,
- parmi deux solutions faisables, celle ayant la meilleure aptitude est conservée,
- parmi deux solutions infaisables, celle possédant la plus petite violation des contraintes est conservée.

L'aptitude d'une solution infaisable ne dépend pas seulement de son niveau de violation de contraintes, mais aussi des autres individus de la population courante, cette formulation ne nécessitant pas l'utilisation de paramètres de pénalité explicites. De plus, une stratégie simple de niche est implémentée dans l'opérateur de sélection par tournois, afin de maintenir la diversité dans la population. Lorsque deux solutions admissibles  $i$  et  $j$  sont comparées, une *distance euclidienne normalisée*  $d_{ij}$  est calculée entre les deux individus. Si cette distance est plus petite qu'une distance critique  $d_{crit}$ , les individus sont comparés en fonction de leurs aptitudes. Sinon, ils ne sont pas comparés et une autre solution  $j$  est testée. De plus, si un nombre prédéfini  $n_f$  de solutions admissibles est testé, et qu'aucune d'entre elles n'est qualifiée en-dessous de la distance critique, alors le  $i^{\text{ème}}$  individu est déclaré comme gagnant. La distance euclidienne normalisée est définie par

$$d_{ij} = \sqrt{\frac{1}{n} \sum_{k=1}^n \left( \frac{x_k^{(i)} - x_k^{(j)}}{x_k^{(u)} - x_k^{(l)}} \right)^2}.$$

Ainsi, les différents individus sont évalués de la manière suivante (voir figure 2.14) : les surfaces des superellipsoïdes sont quadrillées de manière sous-optimale en utilisant la paramétrisation angle-centre  $n$ -D donnée dans la définition 2.5, puis l'aptitude de chaque individu  $F_i$  peut alors être calculée en utilisant l'équation (2.46).

**Remarque 2.10.** De nombreux autres opérateurs de sélection sont codés dans la librairie OCEANS, comme la sélection classique par tournois, l'échantillonnage universel stochastique, la sélection géométrique normalisée, la sélection par roulette et la sélection par reste stochastique sans remplacement.

Dans le domaine de l'optimisation par algorithmes génétiques, un nombre très important d'opérateurs de croisement et de mutation ont été développés. Certains d'entre eux sont adaptés à la plupart des problèmes rencontrés (par exemple le croisement simple/double point, le croisement en ligne intermédiaire/étendu), d'autres servent des problèmes spécifiques. L'opérateur génétique de croisement que nous avons retenu dans le cadre du problème d'annexion superellipsoïdale est le *croisement binaire simulé* (SBX) [157]. Ce dernier est particulièrement adapté à notre problème, car la dispersion des descendants autour des individus parents peut être contrôlée au moyen d'un indice de distribution. En utilisant cet opérateur, toute région contiguë arbitraire peut être explorée, à condition qu'un niveau suffisant de diversité soit maintenu parmi les solutions admissibles parents. Plus précisément, la procédure permettant de calculer deux solutions enfants  $y^{(1)}$  et  $y^{(2)}$ , à partir de deux solutions parents  $x^{(1)}$  et  $x^{(2)}$ , est résumée par l'algorithme 10.

En outre, nous avons retenu dans le cadre de cette étude un *opérateur de mutation paramétrique*, défini dans [158]. Comme pour l'opérateur de croisement SBX, une loi de

**Algorithme 10** : Croisement binaire simulé (SBX).

**Entrées** : Deux solutions parents  $x^{(1)}$  et  $x^{(2)}$ , et un indice de distribution  $\eta_c \geq 0$ .

**Sorties** : Deux solutions enfants  $y^{(1)}$  et  $y^{(2)}$ .

**Algorithme** :

- 1 Tirer d'un nombre aléatoire  $u$  entre 0 et 1.
- 2 Calculer le paramètre  $\bar{\beta}$ , en utilisant une loi de probabilité polynomiale, telle que

$$\bar{\beta} = \begin{cases} (\alpha u)^{\frac{1}{\eta_c+1}} & \text{si } u \leq \frac{1}{\alpha}, \\ \left(\frac{1}{2 - \alpha u}\right)^{\frac{1}{\eta_c+1}} & \text{sinon,} \end{cases}$$

où  $\alpha = 2 - \beta^{-(\eta_c+1)}$ , et  $\beta$  est défini par

$$\beta = 1 + \frac{2}{|x^{(2)} - x^{(1)}|} \min [\min (x^{(1)}, x^{(2)}) - x^l, x^u - \max (x^{(1)}, x^{(2)})].$$

Le paramètre  $\eta_c$  représente l'indice de distribution pouvant prendre toute valeur non négative. Une petite valeur de  $\eta_c$  permet de créer des descendants éloignés des individus parents, et une grande valeur des descendants proches des parents.

- 3 Déterminer les solutions enfants  $y^{(1)}$  et  $y^{(2)}$  par

$$\begin{aligned} y^{(1)} &= 0.5 [(x^{(1)} + x^{(2)}) - \bar{\beta}|x^{(2)} - x^{(1)}|], \\ y^{(2)} &= 0.5 [(x^{(1)} + x^{(2)}) + \bar{\beta}|x^{(2)} - x^{(1)}|]. \end{aligned}$$

probabilité polynomiale est utilisée afin de créer un descendant  $y$  dans un voisinage d'un individu parent  $x$ . Le processus de mutation est résumé par l'algorithme 11

Enfin, une stratégie de *migration* peut être adoptée afin d'obtenir un algorithme génétique multi-populations : plusieurs sous-populations évoluent alors en parallèle et échangent du matériel génétique à intervalles réguliers. Les topologies courantes de migration implémentées dans l'outil OCEANS sont le réseau complet, en anneau, où par voisinage.

#### 2.4.4 Problème de commande optimale convexe

Lorsqu'une ou plusieurs formes superellipsoïdales ont été déterminées comme solutions du problème d'annexion 2.1, le problème de commande optimale (2.6), paramétré dans l'espace des sorties plates, peut être réécrit de manière que les contraintes non linéaires non convexes de trajectoire, ainsi que la fonctionnelle de coût, soient approximés par des formes géométriques convexes. Alors, en utilisant la fonction d'appartenance à un superellipsoïde  $n$ -D déformé, définie par (2.6), l'objectif est de calculer des trajectoires optimales situées à l'intérieur des formes convexes. Le problème de commande optimale



**Algorithme 11** : Mutation paramétrique.

**Entrées** : Un individu parent  $x$ , et un indice de distribution  $\eta_c \geq 0$ .

**Sorties** : Une solution enfant  $y$ .

**Algorithme** :

- 1 Tirer d'un nombre aléatoire  $u$  entre 0 et 1.
- 2 Calculer le paramètre  $\bar{\delta}$  tel que

$$\bar{\delta} = \begin{cases} [2u + (1 - 2u)(1 - \delta)^{\eta_m + 1}]^{\frac{1}{\eta_m + 1}} - 1 & \text{si } u \leq 0.5, \\ 1 - [2(1 - u) + 2(u - 0.5)(1 - \delta)^{\eta_m + 1}]^{\frac{1}{\eta_m + 1}} & \text{sinon.} \end{cases}$$

où

$$\delta = \min [(x - x^l), (x^u - x)] / (x^u - x^l),$$

et  $\eta_m$  représente l'indice de distribution.

- 3 Calculer la solution enfant  $y$  par

$$y = x + \bar{\delta} \Delta_{\max},$$

où  $\Delta_{\max} = x^u - x^l$  représente la perturbation maximale admissible, choisie de telle sorte qu'aucun descendant ne soit créé en dehors de l'intervalle  $[x^l, x^u]$ .

convexe s'écrit alors :

$$\begin{aligned} \min_{\bar{z}(t)} \quad & \mathcal{C}_0(\psi_x(\bar{z}(t_0)), \psi_u(\bar{z}(t_0)), t_0) + \int_{t_0}^{t_f} \mathcal{C}_t(\psi_x(\bar{z}(t)), \psi_u(\bar{z}(t)), t) dt \quad (2.47) \\ & + \mathcal{C}_f(\psi_x(\bar{z}(t_f)), \psi_u(\bar{z}(t_f)), t_f) \end{aligned}$$

t.q.

$$\begin{aligned} l_0 &\leq A_0 \bar{z}(t_0) &&\leq u_0, \\ l_t &\leq A_t \bar{z}(t) &&\leq u_t, \quad t \in [t_0, t_f], \\ l_f &\leq A_f \bar{z}(t_f) &&\leq u_f, \\ L_0 &\leq c_0(\psi_x(\bar{z}(t_0)), \psi_u(\bar{z}(t_0))) &&\leq U_0, \\ 0 &\leq \mathcal{F}_n^i(\Psi^*, \bar{z}(t)) &&\leq 1, \quad t \in [t_0, t_f], \\ L_f &\leq c_f(\psi_x(\bar{z}(t_f)), \psi_u(\bar{z}(t_f))) &&\leq U_f. \end{aligned}$$

où  $\mathcal{F}_n^i(\Psi^*, \bar{z}(t))$ ,  $i = 1, \dots, n_s$ , représentent les fonctions d'appartenance associées aux formes convexes utilisées. En effet, du fait du nombre limité de forme géométriques atteignables, il n'est pas toujours possible d'approximer l'ensemble des contraintes de trajectoire d'un problème de commande optimale par une seule forme convexe. Dans ce cas, la propriété de convexité de l'OCP (2.47) est toujours vérifiée, car l'intersection d'enveloppes convexes est encore une enveloppe convexe [159].

Il est également nécessaire que les contraintes de bord soient satisfaites, c'est à dire que  $\mathcal{F}_n(\Psi^*, \bar{z}(t_0)) \leq 1$  et  $\mathcal{F}_n(\Psi^*, \bar{z}(t_f)) \leq 1$ . Il est alors possible de vérifier que les points extrêmes de la trajectoires se situent bien à l'intérieur des enveloppes convexes associées aux



contraintes de trajectoire, en calculant les distances radiales euclidiennes en dimension  $n$  correspondantes via (2.43).

**Remarque 2.11.** Pour chaque fonction d'appartenance, i.e., chaque forme superellipsoïdale, le problème d'optimisation (2.47) ne fait intervenir que  $n_{\bar{z}}$  paramètres d'optimisation, au lieu de  $n_{\bar{z}} \times q_v$  pour l'approche polytopique décrite dans [12], où  $q_v$  désigne le nombre de sommets du polytope.

Dans la section suivante, nous donnons quelques résultats préliminaires d'annexion superellipsoïdale pour des domaines admissibles non convexes de dimension 3, obtenus grâce à l'outil OCEANS. Une application plus complète est proposée dans le chapitre suivant (section 3.5), dans le cadre de l'optimisation de trajectoires de rentrée atmosphérique.

### 2.4.5 Résultats préliminaires

La méthode d'annexion superellipsoïdale décrite dans la section précédente a été appliquée avec succès à des problèmes de dimensions variées. Dans les exemples suivants, des formes géométriques superquadriques tri-dimensionnelles ont été choisies afin d'approximer quelques domaines non convexes prédéfinis. La surface des superellipsoïdes est paramétrée de manière sous-optimale via la représentation angle-centre (2.25). Les anomalies  $\theta_1$  et  $\theta_2$  sont discrétisées en 17 points régulièrement espacés dans les intervalles  $[-\pi; \pi[$  et  $[-\pi/2, \pi/2]$  respectivement. Les paramètres de forme  $(\varepsilon_1, \varepsilon_2)$  sont autorisés à varier dans l'intervalle<sup>44</sup>  $]0.1, 2]$ . Les paramètres de réglage de l'algorithme génétique retenus sont décrits ci-dessous.

- 60 individus répartis en 3 sous-populations
- topologie de migration par réseau complet,
- nombre maximum de générations : 100,
- opérateur de sélection par tournois avec domination (2.46) et classification linéaire des individus selon leurs aptitudes,
- opérateur de mutation paramétrique (cf algorithme 11), taux de mutation de 5%, et indice de distribution  $\eta_c = 1$ ,
- opérateur de croisement binaire simulé SBX (cf algorithme 10), taux de croisement de 70%, et indice de distribution  $\eta_m = 1$ ,
- Conflit de générations de 90% (nombre de nouveaux individus créés à chaque génération),
- Taux de réinsertion de 70% (nombre de descendants insérés dans la génération suivante),

En outre, les individus constituant la population initiale ont été choisis comme étant des superellipsoïdes élémentaires, avec des demi-grand axes factices, afin qu'ils se trouvent tous

---

<sup>44</sup>de trop petites valeurs des paramètres de forme (par exemple en-dessous de 0.1), peuvent engendrer des problèmes numériques lors du calcul de la fonction d'appartenance.

à l'intérieur de la forme non convexe initiale.

Quelques résultats sont présentés sur la figure 2.15 : (a) le plus grand superellipsoïde contenu dans un domaine non convexe de type cacahuète, décrit par l'équation implicite

$$D_1 = \{x|x \in \mathbb{R}^3, ((x_1 - 0.9)^2 + x_2^2 + x_3^2 - 1) ((x_1 + 0.9)^2 + x_2^2 + x_3^2 - 1) - 0.3 \leq 0\},$$

(b) la plus grande superquadrique contenue dans un domaine non convexe, en forme de croissant, décrit par

$$D_2 = \{x|x \in \mathbb{R}^3, 4x_1^2 (x_1^2 + x_2^2 + x_3^2 + x_3) + x_2^2 (x_2^2 + x_3^2 - 1) \leq 0\},$$

(c) la plus grande forme convexe contenue à l'intérieur d'une forme géométrique de type toupie, décrite par l'équation implicite

$$D_3 = \{x|x \in \mathbb{R}^3, \left(\sqrt{x_1^2 + x_3^2} - 3\right)^3 + x_2^2 - 1 \leq 0\},$$

(d) le plus grand superellipsoïde contenu dans le domaine non convexe défini par

$$D_4 = \{x|x \in \mathbb{R}^3, x_2^2 + x_3^2 - 0.5 \cos x_1 \cos x_2 - 1 \leq 0\}.$$

Le dernier exemple (d) peut être comparé avec l'approximation intérieure obtenue dans [12], en considérant un polytope possédant 8 sommets et 12 facettes. Dans notre cas, seulement 3 paramètres d'optimisation interviennent dans le problème au lieu de 36 pour l'approche polytopique.

La méthode d'annexion superellipsoïdale par algorithmes génétiques proposée a également été appliquée avec succès à des espaces de plus grandes dimensions (jusqu'à 10).

## 2.5 Planification de trajectoires tolérante aux défauts

Comme nous l'avons mentionné en introduction de ce chapitre, la replanification autonome de trajectoires de rentrée atmosphérique peut devenir un élément essentiel d'une chaîne de guidage, pilotage et de navigation tolérante aux défauts, lorsque les ressources énergétiques disponibles pour stabiliser le véhicule en attitude deviennent insuffisantes. En d'autres termes, lorsque ni la reconfiguration de la loi de pilotage, ni l'adaptation du correcteur de poursuite de trajectoires ne parviennent à garantir un certain niveau de robustesse en stabilité et en performances de la chaîne GNC lors de l'apparition d'une défaillance. La replanification de trajectoires en ligne peut alors permettre d'améliorer la robustesse de l'ensemble du système aux défaillances. On peut alors se poser les questions suivantes :

*Quelles trajectoires de rentrée sont admissibles vis-à-vis de la manoeuvrabilité réduite du véhicule, et vont permettre au minimum de ramener le véhicule sur Terre en toute sécurité, et au mieux de minimiser la déviation par rapport à la trajectoire de référence ? En d'autres termes, quelle est alors la trajectoire optimal de vol ?*

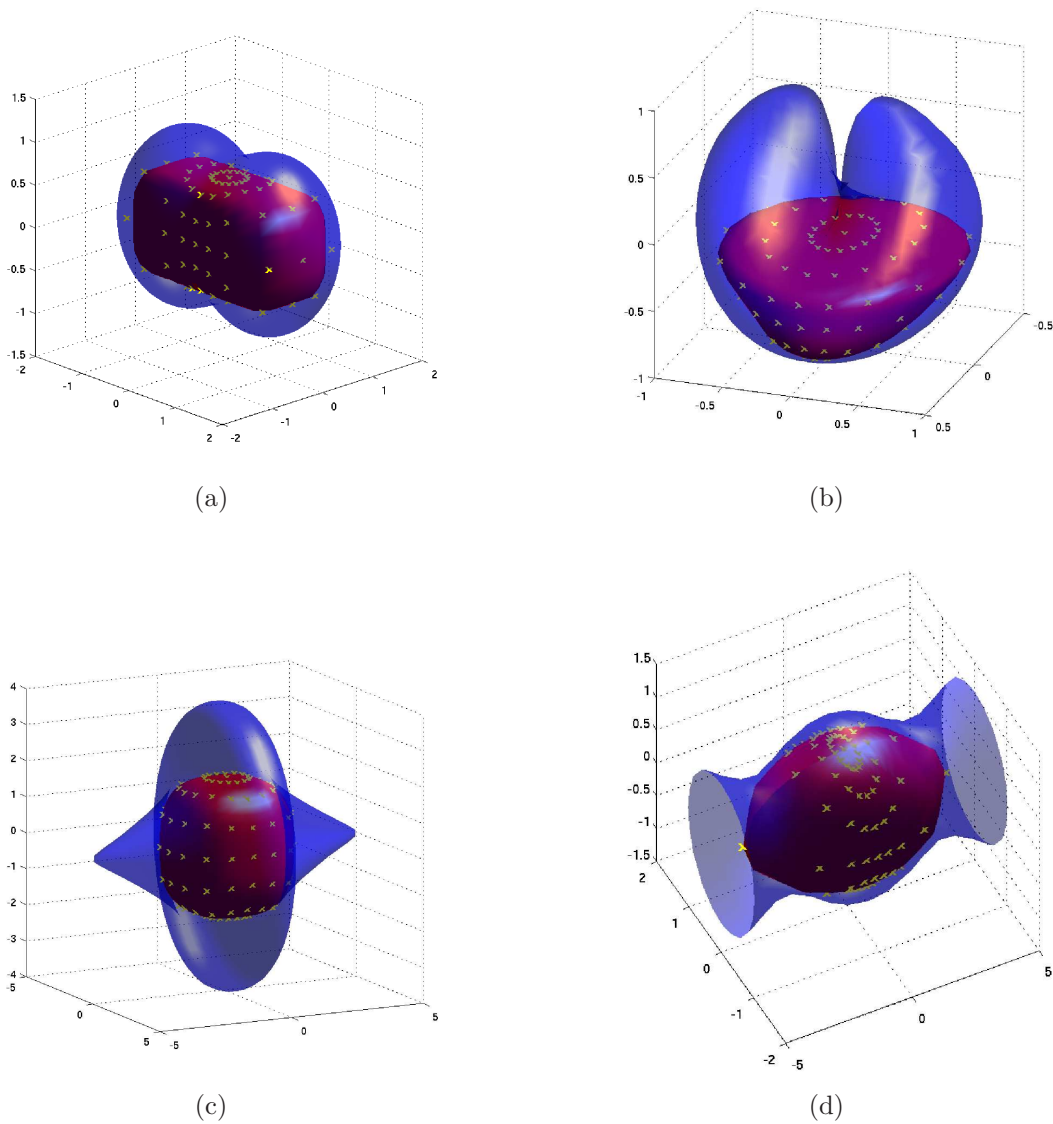


FIG. 2.15 – illustration du processus d'annexion superellipsoïdale.

Nous allons tenter d'y répondre en effectuant une analyse des capacités d'équilibrage statique en rotation du véhicule en situation défaillante. Les cartes de trimmabilité permettent de caractériser l'enveloppe de vol atteignable du véhicule en tenant compte d'un certain nombre de ressources de commande disponibles (gouvernes et/ou système RCS). En fait, l'une des difficultés liées à la replanification de trajectoires à bord est qu'il est nécessaire de prédire l'effet des défauts sur la dynamique du système pour l'ensemble des conditions de vol futures. En outre, les coefficients aérodynamiques du véhicule peuvent changer de manière significative lors de l'apparition d'un défaut, et ce d'autant plus que la surface des gouvernes est grande par rapport à la surface alaire. Il est alors nécessaire d'inclure la nouvelle configuration aérodynamique du véhicule dans l'algorithme de re-

planification de trajectoires. Nous proposons deux solutions alternatives au problème de replanification de trajectoires à bord :

1. la *première solution* fait intervenir de manière explicite les contraintes liées à l'équilibrage statique du véhicule dans le problème de commande optimale, formulé dans l'espace des sorties plates. Nous obtenons alors un nouvel OCP d'ordre plus élevé, dans lequel les nouvelles variables d'optimisation sont les sorties plates et les braquages des surface de contrôle aérodynamique. Cette formulation, permettant de traiter un nombre important de situations défaillantes (défauts simples et multiples de gouvernes), vise à réduire l'effort de modélisation et d'analyse réalisé au sol.
2. la *seconde solution* consiste à "piloter" le problème de commande optimale, transcrit dans l'espace des sorties plates, par un indicateur exogène lié à la trimmabilité statique du véhicule en rotation. Nous obtenons donc un nouveau problème de commande optimale où les variables d'optimisation sont toujours les sorties plates du modèle, mais dans lequel les caractéristiques aérodynamiques, ainsi que le corridor Mach- $\alpha$  lié à l'équilibrage statique du véhicule, sont modifiés en fonction d'un vecteur de paramètres exogène. Ce vecteur est constitué de la configuration des surfaces de contrôle aérodynamique en présence d'un ou plusieurs défauts actionneurs. L'effort de modélisation hors ligne est certes plus important qu'avec la première solution, mais l'OCP obtenu ne requiert pas plus de ressources CPU afin d'être résolu.

Dans le champ couvert par les travaux liés à la commande tolérante aux fautes, un certain nombre de travaux récents se sont focalisés sur la replanification de trajectoires de rentrée atmosphérique à bord. Dans [37], des cartes de trimmabilité statique sont modélisées au sol pour un ensemble de situations défaillantes prédéfinies, et sont ensuite intégrées dans l'algorithme de planification de trajectoires. Le problème de commande optimale obtenu est résolu en utilisant une méthode pseudo-spectrale [160]. Dans [36], les coefficients aérodynamiques sont modélisés de façon à varier dans un corridor dont la géométrie dépend du type de défaut rencontré. La trajectoire TAEM est ensuite paramétrée par un petit nombre de paramètres de dimensionnement qui sont optimisés en ligne par une technique indirecte de tirs simples. La méthode proposée est brièvement décrite dans la section 1.5.2. Un algorithme de replanification de trajectoires basé sur les réseaux de neurones a été développé dans [35] pour la phase d'approche finale et d'atterrissage automatique du X-40. Une importante base de données de trajectoires est d'abord générée hors ligne, en tenant compte de conditions cinématiques initiales variées à l'interface A&L et d'un certain nombre de configurations de gouvernes en situations défaillantes. Les paramètres critiques de dimensionnement des trajectoires sont ensuite encodés au sol par des réseaux de neurones, qui sont ensuite interpolés à bord du véhicule afin de trouver la trajectoire la plus adaptée à la situation défaillante rencontrée. Les coefficients de dimensionnement de la trajectoire interpolée sont alors légèrement modifiés en ligne afin d'éliminer les erreurs sur les conditions de bord. Finalement, A. Trent *et al.* [38] a proposé une méthode de replanification de trajectoires à bord, basée sur une utilisation conjointe de cartes de trimmabilité statique et d'une méthode de tirs indirects modifiée.

### 2.5.1 Cartes de trimmabilité statique

Nous rappelons dans cette section l'algorithme permettant de générer, hors ligne, des cartes d'équilibrage statique, ainsi que les profils des coefficients de portance et de traînée en situation défailante. La méthode associée a été initialement développée dans [161], puis utilisée pour des applications de replanification de trajectoires dans [37, 38]. Nous décrivons ici les grandes lignes de la démarche ; son application à une mission de rentrée dans l'atmosphère de la navette spatiale américaine sera présentée en détails dans le chapitre 3.

Supposons que l'on se place dans des conditions de vol symétriques, c'est à dire à dérapage nul ( $\beta = 0$ ). Cette hypothèse est en adéquation avec les spécifications associées à la planification de trajectoires de rentrée atmosphérique, car on cherche autant que possible à générer des trajectoires équilibrées, offrant ainsi à loi de pilotage un degré de liberté supplémentaire sur la dynamique latérale pour contrer, par exemple, des perturbations aérologiques telles que des rafales de vent, où des défauts d'équilibrage inhérents à la configuration inertielle de l'appareil. Par conséquent, les moments de roulis  $L$  et de dérapage  $N$  seront considérés comme étant identiquement nuls, l'équilibrage en rotation du véhicule étant réalisé uniquement dans un plan vertical.

Nous considérons que l'on dispose d'une base aérodynamique complète du véhicule de rentrée atmosphérique, c'est à dire comprenant l'ensemble des coefficients de forces et de moments de l'appareil. Nous noterons  $C_{m_{0i,j}}$  le coefficient de moment de tangage en configuration lisse, évalué à l'incidence  $\alpha_i$  et au nombre de Mach  $M_j$ , tel que

$$C_{m_{0i,j}} = C_{m_{0i,j}}(\alpha_i, M_j).$$

De plus, comme seul le mouvement longitudinal du véhicule est considéré, nous supposons que les coefficients de moment de roulis et de lacet en configuration lisse sont nuls, i.e.,

$$\begin{aligned} C_{l_{0i,j}}(\alpha_i, M_j) &= 0, \\ C_{n_{0i,j}}(\alpha_i, M_j) &= 0, \end{aligned}$$

Une fois les coefficients de moments en configuration lisse connus, un algorithme d'allocation est utilisé afin de fournir les valeurs des braquages de gouvernes  $\delta_{i,j} \in \mathbb{R}^p$ , où  $p$  représente le nombre de surfaces de contrôle aérodynamique, permettant d'équilibrer le véhicule statiquement en rotation. Ainsi, pour chaque point de l'enveloppe Mach- $\alpha$ , nous cherchons à déterminer un vecteur  $\delta_{i,j}$  tel que

$$\begin{pmatrix} C_{l_{\delta_{i,j}}}(\alpha_i, M_j, \delta_{i,j}) \\ C_{m_{\delta_{i,j}}}(\alpha_i, M_j, \delta_{i,j}) \\ C_{n_{\delta_{i,j}}}(\alpha_i, M_j, \delta_{i,j}) \end{pmatrix} = \begin{pmatrix} 0 \\ -C_{m_{0i,j}}(\alpha_i, M_j) \\ 0 \end{pmatrix},$$

où  $C_{l_{\delta_{i,j}}}(\alpha_i, M_j, \delta_{i,j})$ ,  $C_{m_{\delta_{i,j}}}(\alpha_i, M_j, \delta_{i,j})$  et  $C_{n_{\delta_{i,j}}}(\alpha_i, M_j, \delta_{i,j})$  représentent respectivement les coefficients de roulis, de tangage et de lacet produits par les braquages des gouvernes.

Les débattements des gouvernes sont limités, tels que

$$\underline{\delta} \leq \delta_{i,j} \leq \bar{\delta},$$

où  $\underline{\delta}$  et  $\bar{\delta}$  sont des vecteurs contenant respectivement les valeurs minimales et maximales des braquages. Un défaut de type blocage ou départ en butée sur la gouverne  $\delta_k$  pourra alors être modélisé de manière simple, en posant  $\underline{\delta}_k = \bar{\delta}_k$ , et un défaut engendrant une perte totale d'efficacité de l'élément mobile (gouverne flottante) sera alors caractérisé par une incapacité à produire un moment, i.e.,  $C_{l_{\delta_k}} = C_{m_{\delta_k}} = C_{n_{\delta_k}} = 0$ .

On peut alors poser le problème d'équilibrage statique suivant.

**Problème 2.2** (problème d'équilibrage statique). *Soit  $\delta$  le vecteur de braquages de gouvernes associé aux coefficients de moment de roulis, de tangage et de lacet définis respectivement par  $C_{l_\delta}(\alpha, M, \delta)$ ,  $C_{m_\delta}(\alpha, M, \delta)$  et  $C_{n_\delta}(\alpha, M, \delta)$ . Le coefficient de moment de tangage en configuration lisse est noté  $C_{m_0}(\alpha, M)$ . Le problème d'équilibrage statique est alors défini par le problème de faisabilité :*

$$\min_{\underline{\delta}} J_D = \min_{\underline{\delta}} \left\| \begin{bmatrix} C_{l_{\delta_{i,j}}}(\alpha_i, M_j, \delta_{i,j}) \\ C_{m_{\delta_{i,j}}}(\alpha_i, M_j, \delta_{i,j}) \\ C_{n_{\delta_{i,j}}}(\alpha_i, M_j, \delta_{i,j}) \end{bmatrix} - \begin{bmatrix} 0 \\ -C_{m_{0_{i,j}}}(\alpha_i, M_j) \\ 0 \end{bmatrix} \right\|_1 \quad (2.48)$$

t.q.

$$\underline{\delta} \leq \delta \leq \bar{\delta},$$

en chaque point  $(\alpha_i, M_j)$  de la base aérodynamique.

Le problème (2.48) peut être résolu en utilisant différentes techniques d'allocation reposant sur la résolution de problèmes de programmation linéaire (LP) [162], quadratique (QP) [163], linéaire entier mixte (MILP) [164] ou encore de programmation semi-définie positive (SDP) [165]. Le lecteur intéressé par cette thématique trouvera un état de l'art assez complet des techniques d'allocation existantes dans l'article [162]. Bien que les méthodes de collocation fondées sur la résolution de problèmes de programmation linéaire soient de loin les plus efficaces d'un point de vue numérique, leur utilisation nécessite que les coefficients de moments aérodynamiques soient approximés linéairement autour de positions de gouvernes préférentielles (correspondant souvent à la configuration lisse de l'appareil). Or, les coefficients de moments possèdent pour la plupart des non linéarités plus ou moins fortes associées à des valeurs de braquage importantes des gouvernes. Ce point ne peut être ignoré dans le cadre de la replanification de trajectoires à bord<sup>45</sup>, où les ressources énergétiques limitées engendrent souvent de forts braquages des gouvernes disponibles après l'apparition d'un défaut. Pour palier à ce problème, les coefficients de moment peuvent alors être modélisés au moyen de polynômes par morceaux, et intégrés dans un problème d'optimisation de type MILP [164].

Dans notre cas, nous avons choisi de modéliser les coefficients de moment de manière précise sous forme analytique grâce à des réseaux de neurones (perceptrons multi-couches), et de poser le problème d'équilibrage 2.2 comme un problème de programmation non linéaire (NLP). Ce choix est en adéquation avec l'utilisation qui en sera faite dans les sections 2.5.2 et 2.5.3. En effet, nous chercherons dans la section 2.5.2 à inclure explicitement les

<sup>45</sup>et de la reconfiguration de lois de pilotage.



contraintes de trimmabilité statique du véhicule dans le problème de commande optimale associé à la replanification de trajectoires à bord. Dans un second temps, nous modéliserons les cartes de trimmabilité statique hors ligne, puis nous “piloterons” ensuite l’OCP grâce à un ensemble de paramètres exogènes issus de cette analyse. L’étude étant menée cette fois-ci au sol, il n’existe aucune restriction quant à la stratégie retenue.

Maintenant, considérons que le processus d’allocation renvoie une valeur de braquage  $\delta_{i,j}^*$ , associée à l’équation (2.48), pour un couple de points  $(\alpha_i, M_j)$ . Si  $\delta_{i,j}^*$  est nul, alors des ressources suffisantes sont disponibles afin d’équilibrer longitudinalement le véhicule. Dans le cas contraire, si l’équation (2.48) n’est pas satisfaite, alors le scalaire  $J_D(\delta^*)$  est appelé *déficience de trimmabilité statique*, ou *déficience d’équilibrage*. En réalisant ce test en chaque point Mach- $\alpha$  d’une grille de valeurs prédéfinie, il est alors possible de construire une *carte de trimmabilité statique*, également appelée *carte d’équilibrage statique*. Cette carte indique les zones dans lesquelles le véhicule est équilibrable statiquement dans un plan vertical.

On peut alors déterminer, pour l’ensemble des valeurs du nombre de Mach, le domaine d’incidences permettant d’équilibrer longitudinalement le véhicule. Ce domaine d’incidences peut varier de manière significative en fonction du type et de l’amplitude du défaut considéré. La plage de valeurs d’incidences atteignable constitue alors une nouvelle contrainte du problème de planification de trajectoires. Plus précisément, on cherche alors à déterminer une nouvelle trajectoire permettant au véhicule de poursuivre sa mission, tout en assurant que les incidences de vol sont compatibles avec les capacités d’équilibrage longitudinal du véhicule en situation défaillante.

La figure 2.16 présente un exemple de carte d’équilibrage statique tri-dimensionnelle et sa projection dans le plan Mach- $\alpha$ . Le bandeau latéral représente alors les valeurs des déficiences associées. Cette carte a été obtenue à partir de la base aérodynamique de la navette spatiale américaine STS-1 [166], en considérant une panne simple (élevon intérieur droit bloqué en position à  $20^\circ$ ). Un autre exemple de cartes de trimmabilité est présenté sur la figure 2.17, en considérant un défaut multiple de gouvernes (élevon intérieur droit bloqué à  $20^\circ$ , élevon extérieur droit bloqué à  $-20^\circ$ , et aérofrein bloqué en position nulle). Des résultats plus précis concernant la détermination des cartes d’équilibrage statique seront donnés dans le chapitre 3.

**Remarque 2.12.** La carte de trimmabilité statique obtenue peut être vue comme la projection dans l’espace Mach- $\alpha$  d’un polytope appelé AMS (Attainable Moment Subset) [167, 168, 169], contenant l’ensemble des moments atteignables en fonction d’une configuration aérodynamique donnée  $\underline{\delta} \leq \delta_{i,j} \leq \bar{\delta}$ .

**Remarque 2.13.** Dans le cadre de la replanification autonome de trajectoires, nous cherchons seulement à résoudre le problème de faisabilité défini par 2.2. Si le véhicule est équilibrable longitudinalement, c’est à dire qu’il existe un vecteur de braquages de gouvernes vérifiant l’équation (2.48), tout excès d’énergie peut être utilisé afin de manoeuvrer le véhicule. Alors, les techniques d’allocation développées pour les lois de pilotage comprennent généralement un second problème d’optimisation visant à minimiser l’écart par

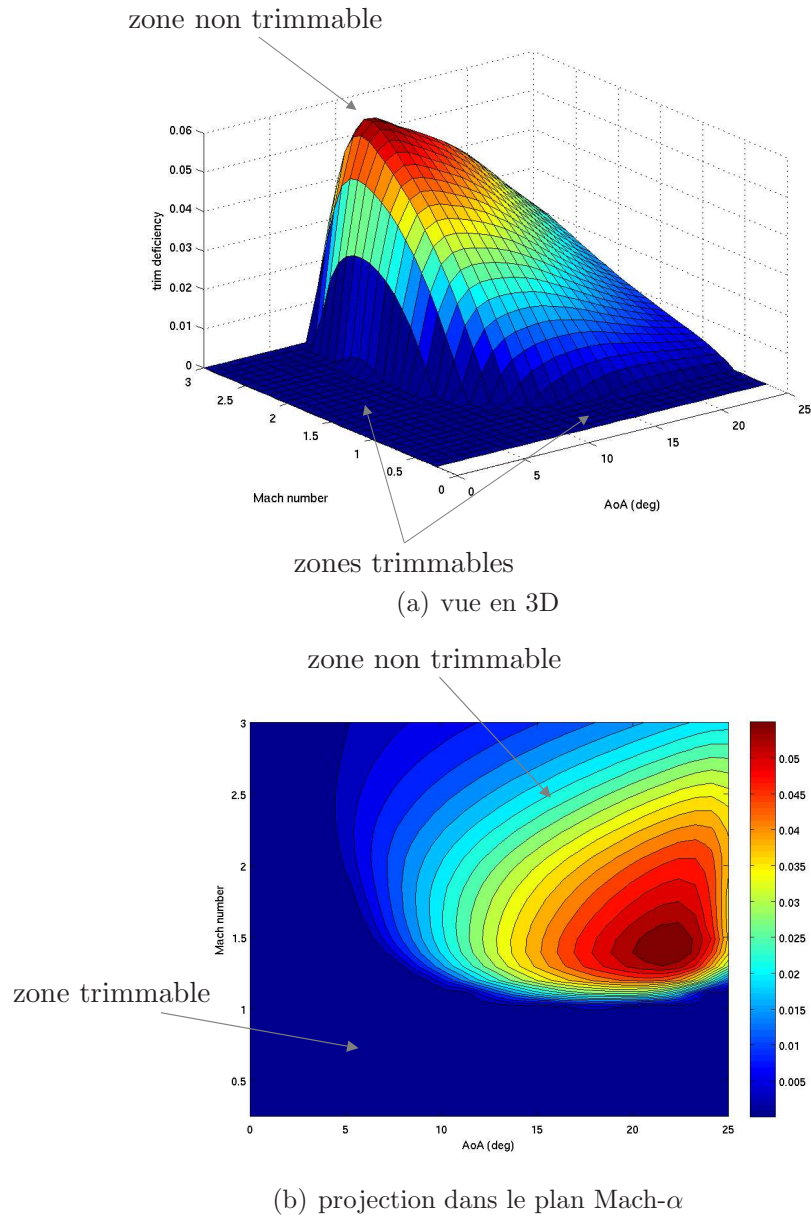


FIG. 2.16 – exemple de carte d'équilibrage statique.

rapport à un vecteur de positions préférées  $\delta_p$ , défini par

$$\min_{\delta} J_S = \min_{\delta} \|W(\delta - \delta_p)\|_1$$

t.q.

$$\underline{\delta} \leq \delta \leq \bar{\delta}$$

et

$$\begin{pmatrix} C_{l_\delta}(\alpha, M, \delta) \\ C_{m_\delta}(\alpha, M, \delta) \\ C_{n_\delta}(\alpha, M, \delta) \end{pmatrix} = \begin{pmatrix} 0 \\ -C_{m_0}(\alpha, M) \\ 0 \end{pmatrix},$$



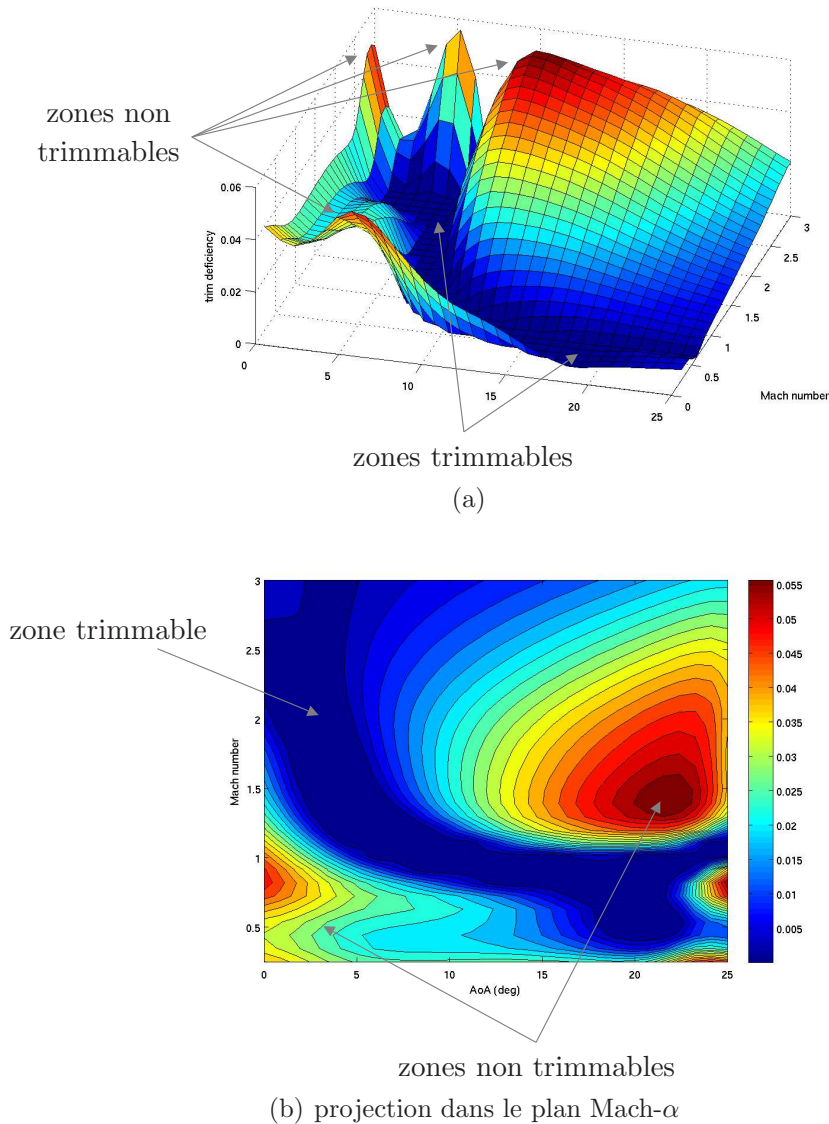


FIG. 2.17 – autre exemple de carte d'équilibrage statique.

où  $W$  est une matrice de pondération permettant d'indiquer l'importance avec laquelle un actionneur doit regagner une position préférentielle.

**Remarque 2.14.** En situation nominale, toutes les cartes d'équilibrage statique sont normalement entièrement atteignables, c'est à dire, en utilisant le code de couleurs précédent, que les cartes représentées sur les figures 2.16 et 2.17 sont complètement bleues.

De la même façon qu'il est possible de générer une carte de trimmabilité statique, il est possible de modéliser les coefficients de forces aérodynamiques résultant d'une situation défailante en chaque point de l'enveloppe de vol atteignable. Ceci peut être réalisé en substituant la solution  $\delta_{i,j}^*$  du problème d'équilibrage 2.2 dans les expressions des coefficients de moment et de traînée du véhicule. Les coefficients totaux de portance et de traînée,

en un point  $(\alpha_i, M_j)$  du plan Mach- $\alpha$  et pour un vecteur  $\delta_{i,j}^*$ , sont exprimés comme la somme des coefficients en configuration lisse et des coefficients dépendant des surfaces de contrôle aérodynamique, tel que

$$\begin{aligned} C_L(\alpha_i, M_j) &= C_{L_0}(\alpha_i, M_j) + C_{L_{\delta_{i,j}^*}}(\alpha_i, M_j, \delta_{i,j}^*), \\ C_D(\alpha_i, M_j) &= C_{D_0}(\alpha_i, M_j) + C_{D_{\delta_{i,j}^*}}(\alpha_i, M_j, \delta_{i,j}^*), \end{aligned}$$

où  $C_L(\alpha_i, M_j)$  et  $C_D(\alpha_i, M_j)$  désignent les coefficients totaux de portance et de traînée,  $C_{L_0}(\alpha_i, M_j)$  et  $C_{D_0}(\alpha_i, M_j)$  représentent respectivement les coefficients de portance et de traînée en configuration lisse, et  $C_{L_{\delta_{i,j}^*}}(\alpha_i, M_j, \delta_{i,j}^*)$  et  $C_{D_{\delta_{i,j}^*}}(\alpha_i, M_j, \delta_{i,j}^*)$  sont les coefficients de portance et de traînée produits par les surfaces de contrôle aérodynamique.

La figure 2.18 illustre la variation du coefficient de portance de la navette spatiale américaine STS-1, due à une défaillance multiple de gouvernes (élevon intérieur droit bloqué à  $20^\circ$ , élevons extérieurs droit et gauche bloqués à  $20^\circ$ ). La figure 2.19 montre l'effet d'une

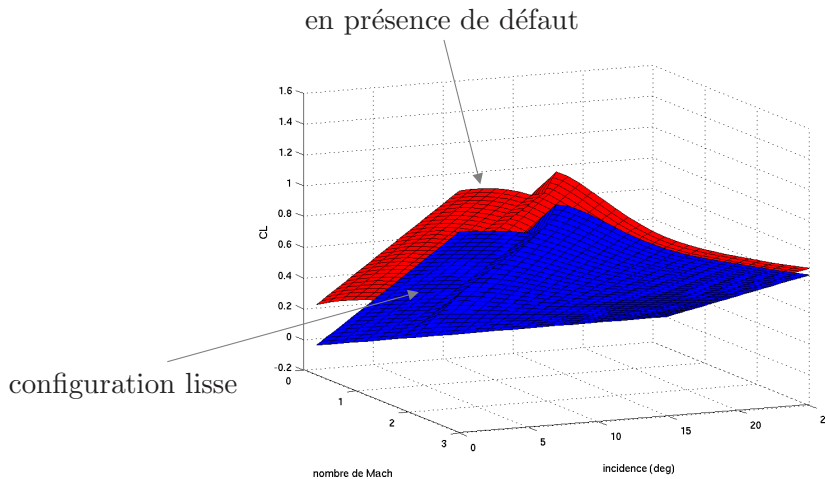


FIG. 2.18 – illustration du coefficient de portance en situation nominale et défaillante.

défaillance multiple de gouvernes sur le coefficient de traînée (élevon intérieur droit bloqué à  $20^\circ$ , élevons extérieurs droit et gauche bloqués à  $20^\circ$ ). Les nouveaux coefficients de portance et de traînée résultant d'une situation défaillante peuvent ensuite être introduits dans le problème de planification de trajectoires afin d'améliorer la précision du modèle dégradé du véhicule en présence de défauts.

Finalement, la démarche présentée dans cette section est résumée de manière plus compacte dans l'algorithme 12. Il est ainsi possible de définir plusieurs niveaux de reconfiguration, résumés dans le tableau 2.1.

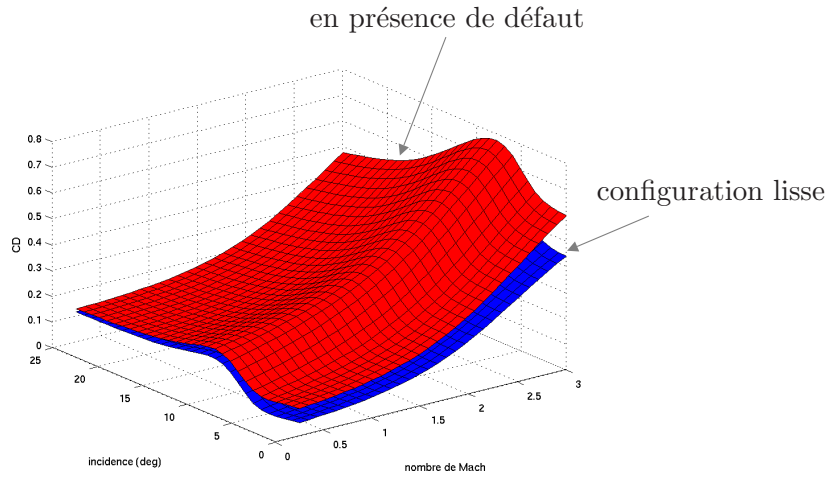


FIG. 2.19 – illustration du coefficient de traînée en situation nominale et défailante.

**Algorithme 12** : Génération d'une carte d'équilibrage statique.

**Entrées** : Vecteur d'incidences de dimension  $n$ , vecteur de nombres de Mach de dimension  $m$ .

**Sorties** : Déficiences d'équilibrage statique  $\text{def}(i, j)$  couvrant l'espace Mach- $\alpha$ .

**Algorithme** :

- 1 Définir une grille de nombres de Mach et de valeurs d'incidences, incluant les bornes inférieure, supérieure, et le pas.
- 2 **pour**  $i = 1$  à  $n$  **faire**
- 3     **pour**  $j = 1$  à  $m$  **faire**
- 4         Calculer le coefficient de moment  $C_{m_{0,i,j}}(\alpha_i, M_j)$  en configuration lisse.
- 5         Résoudre le problème d'équilibrage statique 2.2 pour déterminer le vecteur de braquages de gouvernes  $\delta_{i,j}^*$ .
- 6         Calculer la valeur de déficience d'équilibrage statique définie par

$$\text{def}(i, j) = \left\| \begin{bmatrix} C_{l_{\delta_{i,j}}}(\alpha_i, M_j, \delta_{i,j}) \\ C_{m_{\delta_{i,j}}}(\alpha_i, M_j, \delta_{i,j}) \\ C_{n_{\delta_{i,j}}}(\alpha_i, M_j, \delta_{i,j}) \end{bmatrix} - \begin{bmatrix} 0 \\ -C_{m_{0,i,j}}(\alpha_i, M_j) \\ 0 \end{bmatrix} \right\|_2.$$

Si  $\text{def}(i, j) = 0$ , alors le véhicule peut être équilibré longitudinalement au point  $(\alpha_i, M_j)$ .

- 7         Calculer les coefficients de portance et de traînée au point  $(\alpha_i, M_j)$  en substituant le vecteur  $\delta_{i,j}^*$  dans la base aérodynamique.
- 8     **fin**
- 9 **fin**

Niveaux de criticité	Méthode utilisée
1	Sur la trajectoire de vol courante (qui vérifie la dynamique du véhicule et les contraintes de trajectoire), il suffit de vérifier les contraintes d'équilibrage précédentes en plusieurs points de la trajectoire. Si le véhicule est équilibrable, on peut alors poursuivre la mission en <b>mode nominal</b> . Dans ce cas, un système d'allocation de gouvernes tolérant est utilisé pour reconfigurer la loi de pilotage, et il n'est pas nécessaire de replanifier une nouvelle trajectoire de vol.
2	En situation défaillante, les contraintes de trimmabilité statique ne sont pas vérifiées sur la trajectoire courante, qui n'est donc plus réalisable. En revanche, il existe une autre trajectoire admissible dans l'enveloppe de vol réduite. On poursuit alors la mission en <b>mode dégradé</b> , en replanifiant une nouvelle trajectoire de vol.
3	Il n'existe pas de nouvelle trajectoire permettant de poursuivre la mission mais on peut trouver une trajectoire admissible en modifiant les objectifs de mission, notamment le point visé. On passe alors en mode d' <b>abandon de mission</b> , et la trajectoire replanifiée est une trajectoire de secours permettant uniquement au véhicule de se poser sur une piste de manière sûre.
4	Il n'existe plus de solution envisageable. Nous sommes donc en situation de <b>perte totale du véhicule</b> .

TAB. 2.1 – niveaux de criticité associés à la replanification de trajectoires à bord.

## 2.5.2 Première solution : prise en compte directe des contraintes d'équilibrage dans le problème de commande optimale

Dans cette section, nous présentons la première solution retenue dans le cadre de la replanification de trajectoires de rentrée atmosphérique à bord. L'idée consiste à intégrer les contraintes d'équilibrage statique (2.48) dans le problème de commande optimale défini par (2.6) dans l'espace des sorties plates. Les contraintes d'équilibrage sont alors uniquement évaluées aux points de collocation définis dans le problème de commande optimale initial. En effet, le but n'est pas ici de développer un module d'allocation, pour lequel des vecteurs de braquages de gouvernes sont généralement calculés à la même fréquence que la loi de pilotage<sup>46</sup>, mais plutôt de vérifier que l'on reste à l'intérieur d'un corridor atteignable dans l'espace Mach- $\alpha$  le long de la trajectoire générée, défini par des déficiences d'équilibrage statique nulles.

En fait, le nouveau problème de commande optimale fait intervenir  $p$  variables d'op-

<sup>46</sup>de manière approximative, on peut considérer que la boucle de pilotage est rafraîchie à une fréquence au moins dix fois plus grande que celle de la boucle de guidage afin d'assurer un découplage fréquentiel des deux boucles.

timisation supplémentaires, correspondant aux positions des gouvernes. En effet, nous verrons dans le chapitre 3 que, de par la propriété de platitude du modèle à trois degrés de liberté du véhicule, l'angle d'incidence  $\alpha$  et le nombre de Mach  $M$  peuvent s'exprimer exclusivement en fonction des sorties plates et de leurs dérivées, notées  $\bar{z}$  dans l'espace des jets d'ordre infini. En d'autres termes, il est possible de réécrire les contraintes de trimmabilité statique (2.48) en fonction de  $\bar{z}$  et  $\delta$  sous la forme

$$\begin{pmatrix} C_{l_\delta}(\psi_\alpha(\bar{z}), \psi_M(\bar{z}), \delta) \\ C_{m_\delta}(\psi_\alpha(\bar{z}), \psi_M(\bar{z}), \delta) \\ C_{n_\delta}(\psi_\alpha(\bar{z}), \psi_M(\bar{z}), \delta) \end{pmatrix} = \begin{pmatrix} 0 \\ -C_{m_0}(\psi_\alpha(\bar{z}), \psi_M(\bar{z})) \\ 0 \end{pmatrix}, \quad (2.49)$$

où  $C_{m_0}(\psi_\alpha(\bar{z}), \psi_M(\bar{z}))$  représente le coefficient de moment de tangage en configuration lisse, et où  $C_{l_\delta}(\psi_\alpha(\bar{z}), \psi_M(\bar{z}), \delta)$ ,  $C_{m_\delta}(\psi_\alpha(\bar{z}), \psi_M(\bar{z}), \delta)$  et  $C_{n_\delta}(\psi_\alpha(\bar{z}), \psi_M(\bar{z}), \delta)$  représentent respectivement les coefficients de roulis, de tangage et de lacet produits par les gouvernes.

De même que pour les coefficients de forces, les coefficients de moments aérodynamiques doivent être approximés sous la forme de fonctions analytiques différentiables afin de pouvoir utiliser le concept de platitude différentielle. Nous verrons dans la section 3.3 du chapitre 3 qu'il est possible d'approximer ces coefficients en utilisant soit une analyse en composantes principales, soit des réseaux de neurones statiques, où encore une combinaison des deux solutions précédentes, le but étant bien sûr d'obtenir des modèles non linéaires précis et dépendant d'un nombre réduit de paramètres.

De plus, les coefficients totaux de portance et de traînée, définis en fonction du vecteur  $\bar{z}$ , sont intégrés dans l'OCP (2.6) à la place des coefficients en configuration lisse, tels que

$$C_L(\psi_\alpha(\bar{z}), \psi_M(\bar{z})) = C_{L_0}(\psi_\alpha(\bar{z}), \psi_M(\bar{z})) + C_{L_\delta}(\psi_\alpha(\bar{z}), \psi_M(\bar{z}), \delta), \quad (2.50)$$

$$C_D(\psi_\alpha(\bar{z}), \psi_M(\bar{z})) = C_{D_0}(\psi_\alpha(\bar{z}), \psi_M(\bar{z})) + C_{D_\delta}(\psi_\alpha(\bar{z}), \psi_M(\bar{z}), \delta), \quad (2.51)$$

où  $C_L(\psi_\alpha(\bar{z}), \psi_M(\bar{z}))$  et  $C_D(\psi_\alpha(\bar{z}), \psi_M(\bar{z}))$  représentent les coefficients totaux de portance et de traînée,  $C_{L_0}(\psi_\alpha(\bar{z}), \psi_M(\bar{z}))$  et  $C_{D_0}(\psi_\alpha(\bar{z}), \psi_M(\bar{z}))$  les coefficients de portance et de traînée en configuration lisse, et  $C_{L_\delta}(\psi_\alpha(\bar{z}), \psi_M(\bar{z}), \delta)$  et  $C_{D_\delta}(\psi_\alpha(\bar{z}), \psi_M(\bar{z}), \delta)$  les coefficients de portance et de traînée produits par les surfaces de contrôle aérodynamique  $\delta$ . Nous décrirons également dans le chapitre 3 la technique retenue pour modéliser les coefficients de forces sous forme analytique.

Maintenant, en introduisant les nouvelles contraintes de trajectoire (2.49), (2.50) et (2.51)

dans le problème de commande optimale (2.6), on obtient alors un nouvel OCP qui s'écrit

$$\begin{aligned} \min_{\bar{z}(t), \delta(t)} \quad & \mathcal{C}_0(\psi_x(\bar{z}(t_0)), \psi_u(\bar{z}(t_0), \delta(t_0)), t_0) + \int_{t_0}^{t_f} \mathcal{C}_t(\psi_x(\bar{z}(t)), \psi_u(\bar{z}(t), \delta(t)), t) dt \\ & + \mathcal{C}_f(\psi_x(\bar{z}(t_f)), \psi_u(\bar{z}(t_f), \delta(t_f)), t_f) \end{aligned} \quad (2.52)$$

t.q.

$$\begin{aligned} l_0 &\leq A_0 \bar{z}(t_0) \leq u_0, \\ l_t &\leq A_t \bar{z}(t) \leq u_t, \quad t \in [t_0, t_f], \\ l_f &\leq A_f \bar{z}(t_f) \leq u_f, \\ L_0 &\leq c_0(\psi_x(\bar{z}(t_0)), \psi_u(\bar{z}(t_0), \delta(t_0))) \leq U_0, \\ L_t &\leq c_t(\psi_x(\bar{z}(t)), \psi_u(\bar{z}(t), \delta(t))) \leq U_t, \quad t \in [t_0, t_f], \\ L_f &\leq c_f(\psi_x(\bar{z}(t_f)), \psi_u(\bar{z}(t_f), \delta(t_f))) \leq U_f, \end{aligned}$$

et

$$\underline{\delta} \leq \delta(t) \leq \bar{\delta}, \quad t \in [t_0, t_f].$$

En utilisant la méthodologie décrite dans la section 2.4, permettant de convexifier un problème de commande optimale en optimisant des formes superquadriques déformables, il est possible de réaliser la même opération pour l'OCP (2.52). On obtient alors un problème de commande optimale convexe qui s'écrit

$$\begin{aligned} \min_{\bar{z}(t), \delta(t)} \quad & \mathcal{C}_0(\psi_x(\bar{z}(t_0)), \psi_u(\bar{z}(t_0), \delta(t_0)), t_0) + \int_{t_0}^{t_f} \mathcal{C}_t(\psi_x(\bar{z}(t)), \psi_u(\bar{z}(t), \delta(t)), t) dt \\ & + \mathcal{C}_f(\psi_x(\bar{z}(t_f)), \psi_u(\bar{z}(t_f), \delta(t_f)), t_f) \end{aligned} \quad (2.53)$$

t.q.

$$\begin{aligned} l_0 &\leq A_0 \bar{z}(t_0) \leq u_0, \\ l_t &\leq A_t \bar{z}(t) \leq u_t, \quad t \in [t_0, t_f], \\ l_f &\leq A_f \bar{z}(t_f) \leq u_f, \\ L_0 &\leq c_0(\psi_x(\bar{z}(t_0)), \psi_u(\bar{z}(t_0), \delta(t_0))) \leq U_0, \\ 0 &\leq \mathcal{F}_n^i(\Psi^*, \bar{z}(t), \delta(t)) \leq 1, \quad t \in [t_0, t_f], \\ L_f &\leq c_f(\psi_x(\bar{z}(t_f)), \psi_u(\bar{z}(t_f), \delta(t_f))) \leq U_f, \end{aligned}$$

et

$$\underline{\delta} \leq \delta(t) \leq \bar{\delta}, \quad t \in [t_0, t_f],$$

où  $\mathcal{F}_n^i(\Psi^*, \bar{z}(t), \delta)$ ,  $i = 1, \dots, n_s$ , représentent les fonctions d'appartenance associées aux formes convexes utilisées, de dimension  $(n_{\bar{z}} + p)$ , où  $n_{\bar{z}}$  représente le nombre d'éléments du vecteur  $\bar{z}$ , et  $p$  désigne le nombre de gouvernes dont l'appareil est équipé.

Décrivons maintenant l'interaction du module de replanification de trajectoires à bord avec le reste de du système (voir figure 2.1). Supposons qu'à l'instant  $t_0$ , le véhicule amorce par exemple la phase de dissipation d'énergie en zone terminale (TAEM) en situation nominale. Le véhicule étant équilibrable longitudinalement sur toute l'enveloppe de vol, une trajectoire de rentrée optimale peut alors être générée entre les instants  $t_0$  et  $t_f$ , en résolvant indifféremment le problème de commande optimale (2.6), (2.52) ou (2.53). Supposons maintenant qu'une panne apparaît à un instant  $t_p$ . Cette panne est d'abord

détectée, puis isolée par le module de diagnostic (FDI) dans la couche de supervision (cf figure 2.1). Ce module de détection et de localisation de défaut peut par exemple être réalisé à partir d'un banc de filtres de diagnostic synthétisés au sol [170]. De plus, on suppose qu'un algorithme d'identification de défauts permet de quantifier l'amplitude de la défaillance, c'est à dire la position des gouvernes bloquées en position. Récemment, une stratégie, basée sur les techniques de filtrage non linéaires, a été proposée dans [171] pour réaliser cette tâche. Une fois les positions des gouvernes défaillantes connues, cette information est directement introduite dans le problème de commande optimale (2.52) ou (2.53), en modifiant les bornes  $\underline{\delta}$  et  $\bar{\delta}$  sur les braquages de gouvernes, puis cet OCP est résolu à bord afin de déterminer une trajectoire admissible de  $t_p$  à  $t_f$ . Les contraintes de trimmabilité statique dans l'espace Mach- $\alpha$  permettent alors de garantir que la nouvelle trajectoire planifiée se trouve à l'intérieur de l'enveloppe de vol réduite du véhicule.

Pour clore cette section, donnons les principaux avantages et inconvénients de la méthode de replanification de trajectoires autonome proposée.

**Avantages :** clairement, le faible nombre d'hypothèses posées concernant la nature<sup>47</sup> et l'amplitude des défaillances rencontrées confèrent à l'algorithme de replanification de trajectoires un niveau d'autonomie assez important.

**Inconvénients :** Le nombre  $p$  de variables d'optimisation supplémentaires, correspondant aux nombre de surfaces de contrôle aérodynamiques, fait passer le nombre total de variables d'optimisation du problème de commande optimale (2.52) de  $n_{\bar{z}}$  à  $n_{\bar{z}} + p$ , ce qui se traduit bien sûr en terme de charge de calcul CPU lors de la résolution du problème. Nous verrons dans le chapitre 3.6 qu'il sera possible de diminuer cette charge de calculs supplémentaire en paramétrant les variables d'optimisation associées aux gouvernes par des B-splines d'ordre 0, c'est à dire que les valeurs des gouvernes seront considérées comme constantes entre les points de collocation de l'OCP.

### 2.5.3 Seconde solution : replanification “pilotée” de trajectoires

La seconde solution retenue dans le cadre de la replanification de trajectoires tolérante aux défauts nécessite un effort de modélisation plus important au niveau sol, mais a pour conséquence de ne pas introduire de variables d'optimisation supplémentaires dans le problème de commande optimale associé. Cette approche a également été utilisée dans [37, 38] afin de planifier des trajectoires de rentrée tolérantes aux défauts.

La démarche proposée consiste alors à calculer un nombre important de cartes d'équilibrage statiques, obtenues grâce à l'algorithme 12 pour un sous-ensemble de défauts prédéfinis  $\Delta \triangleq \{\delta_{f_1}, \delta_{f_2}, \dots, \delta_{f_K}\}$  de dimension  $K$ , et à modéliser dans chaque cas les corridors Mach- $\alpha$  admissibles obtenus ( $\alpha(M, \delta_{f_1}), \alpha(M, \delta_{f_2}), \dots, \alpha(M, \delta_{f_K})$ ). Ces corridors sont ensuite introduits dans le problème de replanification de trajectoires à bord.

---

<sup>47</sup>rappelons ici que l'on considère uniquement des défauts simples ou multiples de type blocage ou départ en butée, ainsi qu'une perte totale d'efficacité des surface de contrôle aérodynamique.



On obtient alors un ensemble de contraintes de trajectoires portant sur l'incidence, telles que

$$\alpha_l(M, \delta_g) \leq \alpha(t) \leq \alpha_u(M, \delta_g), \quad t \in [t_0, t_f], \quad \delta_g \in \Delta, \quad (2.54)$$

où  $\alpha_l(M, \delta_g)$  et  $\alpha_u(M, \delta_g)$  représentent respectivement les bornes inférieures et supérieures du corridor atteignable dans le plan Mach- $\alpha$ , et où  $\delta_g$  désigne une configuration défaillante de gouvernes appartenant à un ensemble prédéfini  $\Delta \triangleq \{\delta_{f_1}, \delta_{f_2}, \dots, \delta_{f_K}\}$ .

De plus, en utilisant la propriété de platitude du modèle, l'expression (2.54) peut se réécrire sous la forme

$$\alpha_l(\psi_M(\bar{z}), \delta_g) \leq \psi_\alpha(\bar{z})(t) \leq \alpha_u(\psi_M(\bar{z}), \delta_g), \quad t \in [t_0, t_f], \quad \delta_g \in \Delta, \quad (2.55)$$

qui dépend alors des sorties plates, de leurs dérivées, et du vecteur de paramètres exogène  $\delta_g$ , permettant de recalculer le corridor admissible  $\alpha(M)$  en présence d'une situation défaillante.

De même, il est possible de créer une base de données contenant l'ensemble des coefficients de portance et de traînée totaux pour le sous-ensemble de situations défaillantes représenté par  $\Delta$ . On obtient alors des expressions de la forme

$$\begin{aligned} C_L(\alpha, M, \delta_g) &= C_{L_0}(\alpha, M) + C_{L_{\delta_g}}(\alpha, M, \delta_g), \\ C_D(\alpha, M, \delta_g) &= C_{D_0}(\alpha, M) + C_{D_{\delta_g}}(\alpha, M, \delta_g), \end{aligned}$$

où  $C_L(\alpha, M, \delta_g)$  et  $C_D(\alpha, M, \delta_g)$  désignent les coefficients totaux de portance et de traînée en fonction du défaut  $\delta_g \in \Delta$ ,  $C_{L_0}(\alpha, M)$  et  $C_{D_0}(\alpha, M)$  représentent les coefficients de portance et de traînée en configuration lisse, et  $C_{L_{\delta_g}}(\alpha, M, \delta_g)$  et  $C_{D_{\delta_g}}(\alpha, M, \delta_g)$  sont les coefficients de portance et de traînée produits par les positions des surfaces de contrôle aérodynamique associées à la situation défaillante  $\delta_g$ .

En utilisant la propriété de platitude du modèle, les coefficients  $C_L$  et  $C_D$  peuvent se réécrire sous la forme

$$C_L(\psi_\alpha(\bar{z}), \psi_M(\bar{z}), \delta_g) = C_{L_0}(\psi_\alpha(\bar{z}), \psi_M(\bar{z})) + C_{L_{\delta_g}}(\psi_\alpha(\bar{z}), \psi_M(\bar{z}), \delta_g), \quad (2.56)$$

$$C_D(\psi_\alpha(\bar{z}), \psi_M(\bar{z}), \delta_g) = C_{D_0}(\psi_\alpha(\bar{z}), \psi_M(\bar{z})) + C_{D_{\delta_g}}(\psi_\alpha(\bar{z}), \psi_M(\bar{z}), \delta_g). \quad (2.57)$$

On obtient ainsi des modèles des coefficients de portance et de traînée, paramétrés par l'indicateur  $\delta_g \in \Delta$ , qu'il est alors possible d'intégrer sous cette forme dans le problème de replanification de trajectoires.

En introduisant les nouvelles contraintes de trajectoire (2.55), (2.56) et (2.57) dans le problème de commande optimale défini par (2.6), on obtient un nouvel OCP, dépendant



du vecteur de paramètres exogènes  $\delta_g$ , tel que

$$\min_{\bar{z}(t)} \quad \mathcal{C}_0(\psi_x(\bar{z}(t_0)), \psi_u(\bar{z}(t_0), \delta_g), t_0) + \int_{t_0}^{t_f} \mathcal{C}_t(\psi_x(\bar{z}(t)), \psi_u(\bar{z}(t), \delta_g), t) dt \quad (2.58)$$

$$+ \mathcal{C}_f(\psi_x(\bar{z}(t_f)), \psi_u(\bar{z}(t_f), \delta_g), t_f)$$

t.q.

$$\begin{aligned} l_0 &\leq A_0 \bar{z}(t_0) &&\leq u_0, \\ l_t &\leq A_t \bar{z}(t) &&\leq u_t, \quad t \in [t_0, t_f], \\ l_f &\leq A_f \bar{z}(t_f) &&\leq u_f, \\ L_0 &\leq c_0(\psi_x(\bar{z}(t_0)), \psi_u(\bar{z}(t_0), \delta_g)) &&\leq U_0, \\ L_t &\leq c_t(\psi_x(\bar{z}(t)), \psi_u(\bar{z}(t), \delta_g)) &&\leq U_t, \quad t \in [t_0, t_f], \\ L_f &\leq c_f(\psi_x(\bar{z}(t_f)), \psi_u(\bar{z}(t_f), \delta_g)) &&\leq U_f, \end{aligned}$$

où  $\delta_g \in \Delta = \{\delta_{f_1}, \delta_{f_2}, \dots, \delta_{f_K}\}$  représente un indicateur exogène associé à la situation défaillante et permettant de “piloter” le problème de commande optimale (2.58).

**Remarque 2.15.** Comme nous l’avons indiqué précédemment, les variables d’optimisation intervenant dans le problème de commande optimale (2.58) sont uniquement les sorties plates et leurs dérivées  $\bar{z}$ . Le problème (2.58) ne comporte pas de variables supplémentaires liées aux profils de braquage des surfaces de contrôle aérodynamique.

**Remarque 2.16.** Notons en outre que les contraintes de trajectoire  $\underline{\delta} \leq \delta(t) \leq \bar{\delta}$ , liées aux actionneurs, n’apparaissent plus dans le problème (2.58), ces dernières étant prises en compte lors de la détermination des cartes d’équilibrage statique.

En utilisant à nouveau la démarche d’annexion superellipsoïdale décrite dans la section 2.4, il est possible de convexifier le problème de commande optimale (2.58). On obtient alors un nouvel OCP, défini par

$$\min_{\bar{z}(t)} \quad \mathcal{C}_0(\psi_x(\bar{z}(t_0)), \psi_u(\bar{z}(t_0), \delta_g), t_0) + \int_{t_0}^{t_f} \mathcal{C}_t(\psi_x(\bar{z}(t)), \psi_u(\bar{z}(t), \delta_g), t) dt \quad (2.59)$$

$$+ \mathcal{C}_f(\psi_x(\bar{z}(t_f)), \psi_u(\bar{z}(t_f), \delta_g), t_f)$$

t.q.

$$\begin{aligned} l_0 &\leq A_0 \bar{z}(t_0) &&\leq u_0, \\ l_t &\leq A_t \bar{z}(t) &&\leq u_t, \quad t \in [t_0, t_f], \\ l_f &\leq A_f \bar{z}(t_f) &&\leq u_f, \\ L_0 &\leq c_0(\psi_x(\bar{z}(t_0)), \psi_u(\bar{z}(t_0), \delta_g)) &&\leq U_0, \\ 0 &\leq \mathcal{F}_n^i(\Psi^*, \bar{z}(t), \delta_g) &&\leq 1, \quad t \in [t_0, t_f], \\ L_f &\leq c_f(\psi_x(\bar{z}(t_f)), \psi_u(\bar{z}(t_f), \delta_g)) &&\leq U_f, \end{aligned}$$

où  $\delta_g \in \Delta = \{\delta_{f_1}, \delta_{f_2}, \dots, \delta_{f_K}\}$  est un indicateur exogène associé à la situation défaillante et permettant de “piloter” le problème de commande optimale (2.59). De plus,  $\mathcal{F}_n^i(\Psi^*, \bar{z}(t), \delta_g)$ ,  $i = 1, \dots, n_s$ ,  $\delta_g \in \Delta$ , représentent les fonctions d’appartenance associées aux formes convexes utilisées, de dimension  $(n_{\bar{z}})$ , où  $n_{\bar{z}}$  représente le nombre de sorties plates et leurs dérivées.

Décrivons à présent un scénario typique de replanification de trajectoires à bord en présence d'une situation défaillante. Supposons qu'à l'instant  $t_0$ , le véhicule soit équilibrable longitudinalement sur toute l'enveloppe de vol. Une trajectoire de rentrée optimale peut alors être planifiée entre les instants  $t_0$  et  $t_f$ , en résolvant indifféremment le problème de commande optimale (2.6), (2.58) ou (2.59). Supposons maintenant qu'une défaillance apparaisse à un instant  $t_p$ . Comme précédemment, on suppose que la situation défaillante est détectée, isolée, puis quantifiée grâce à une stratégie de diagnostic couplée à un algorithme d'identification de défauts. On obtient ainsi l'indicateur  $\delta_g$ , qui va permettre, grâce aux bases de données de corridors Mach- $\alpha$  et de coefficients de portance et de trainée générées au sol, de "piloter" le problème de commande optimale (2.58) ou (2.59) en fonction de la situation défaillante rencontrée. Une trajectoire de rentrée est ensuite générée entre les instants  $t_p$  et  $t_f$  à partir de ce nouveau modèle.

Afin de clore cette section, résumons les principaux avantages et inconvénients associés à cette seconde méthode de replanification de trajectoires autonome.

**Avantages :** comme il n'y a pas de variables d'optimisation additionnelles dans le problème de commande optimale (2.58) (ou (2.59)), les ressources CPU nécessaires pour résoudre le problème de replanification de trajectoires tolérant aux défauts seront du même ordre de grandeur que celles nécessaires pour résoudre le problème initial (2.6). Les sorties plates sont alors les seules variables d'optimisation du problème de commande optimale tolérant aux défauts.

**Inconvénients :** la détermination au sol d'une banque de corridors Mach- $\alpha$  admissibles et de coefficients de forces aérodynamiques en présence de défauts nécessite de définir un sous-ensemble de scénarios défaillants représentatifs du problème considéré. Il n'est évidemment pas imaginable de considérer l'ensemble des combinaisons possibles de pannes simples/multiples de gouvernes bloquées ou flottantes pouvant potentiellement survenir en situation défaillante. Ainsi, le problème de commande optimale défini par (2.58) permet de traiter un ensemble de défauts plus réduit qu'en utilisant la formulation générale développée dans la section 2.5.2, donnant lieu au problème de commande optimale (2.52).

## 2.6 Transformation en un problème de programmation non linéaire

La dernière étape du processus de planification de trajectoires consiste à paramétrer les variables d'optimisation par des courbes différentiables, et à discrétiser le problème de commande optimale (2.53) (ou (2.59)) afin de le rendre fini. Cette méthode est connue sous le nom de transcription (ou collocation) directe (voir le chapitre 1 pour plus de détails). La transcription du problème de commande optimale (2.53) (ou (2.59)) en un problème de programmation non linéaire dans l'espace des sorties plates est accomplie

en paramétrant les variables d'optimisation (sorties plates et, le cas échéant<sup>48</sup>, braquages des gouvernes) par des fonctions polynomiales par morceaux, suivie d'une discrétisation équilibrée en fonction du temps.

Un grand nombre de courbes candidates peuvent être utilisées afin de représenter les variables d'optimisation du problème de commande optimale, telles que les polynômes simples, les séries de Fourier, les segments rationnels, etc. En plus d'être capable de représenter de manière précise une base des solutions du problème de planification de trajectoires considéré avec un nombre de variables de décision raisonnable, il est également nécessaire que les courbes soient de classe  $C^k$ , sans qu'elles n'engendrent de contraintes additionnelles dans le problème d'optimisation. La spécification d'un niveau de continuité est nécessaire, car l'état et les commandes du système sont des fonctions des sorties plates et d'un nombre fini de leurs dérivées. La propriété de support local est également souhaitée pour la base de fonctions. Cela signifie que les paramètres de contrôle de la courbe agissent uniquement dans un voisinage du point considéré, le reste de la courbe demeurant inchangé. La propriété de support local permet d'améliorer la stabilité numérique lors de la résolution du problème d'optimisation. Par exemple, un polynôme simple d'ordre relativement élevé serait nécessaire afin de satisfaire des contraintes complexes, et la détermination des coefficients de ce polynôme résulterait en un problème mal conditionné numériquement, donc inefficace.

Une solution permettant de répondre à la plupart des conditions précédentes consiste à paramétrer les variables d'optimisation par des polynômes de Bézier continus par morceaux, communément appelés B-splines. Quelques notions fondamentales concernant les B-splines sont rappelées en annexe B.

En particulier, les sorties plates  $\{z_i\}_{i=1}^m$  peuvent être paramétrées, pour  $t_0 \leq t \leq t_f$ , par des courbes B-splines définies par

$$\begin{aligned} z_1(t, p^1) &= \sum_{i=0}^{q_1} p_j^1 \mathcal{B}_{i,k_1}^{(0)}(t) \quad \text{pour la séquence nodale } \eta_1, \\ z_2(t, p^2) &= \sum_{i=0}^{q_2} p_j^2 \mathcal{B}_{i,k_2}^{(0)}(t) \quad \text{pour la séquence nodale } \eta_2, \\ &\vdots \\ z_m(t, p^m) &= \sum_{i=0}^{q_m} p_j^m \mathcal{B}_{i,k_m}^{(0)}(t) \quad \text{pour la séquence nodale } \eta_m, \end{aligned}$$

où  $\mathcal{B}_{i,k_j}^{(0)}(t)$  est la dérivée d'ordre 0 de la  $i^{\text{ème}}$  fonction de la base de B-splines de degré  $k_j$ , construite sur la séquence nodale  $\eta_j$ , et  $p^j$  est le vecteur de *points de contrôle* associé [172] (voir également l'annexe B).

**Remarque 2.17.** Les autres variables d'optimisation intervenant dans l'OCP (2.53),

---

<sup>48</sup>Lorsque les contraintes de trimmabilité statique sont prises en compte dans le problème de commande optimale.

telles que la position des gouvernes, ou le paramètre permettant de spécifier la durée libre de la trajectoire, peuvent également être paramétrés par des courbes B-splines.

La dernière étape consiste à discrétiser le problème de commande optimale (2.53) ou (2.59), dans lesquels les variables d'optimisation sont paramétrées par des B-splines. A cette fin, considérons une distribution uniforme de l'espace temps  $[t_0, t_f]$ , telle que

$$t_0 = \tau_1 < \tau_2 < \tau_N = t_f,$$

où  $N$  désigne un nombre prédéfini de points de collocation. Rappelons que les *points de collocation* représentent les points de l'intervalle  $[t_0, t_f]$  où les contraintes du problème de commande optimal sont satisfaites (voir figure 2.20). Sans perte de généralité, nous avons supposé qu'une séquence unique de points de collocation est utilisée pour vérifier l'ensemble des contraintes de trajectoire du problème.

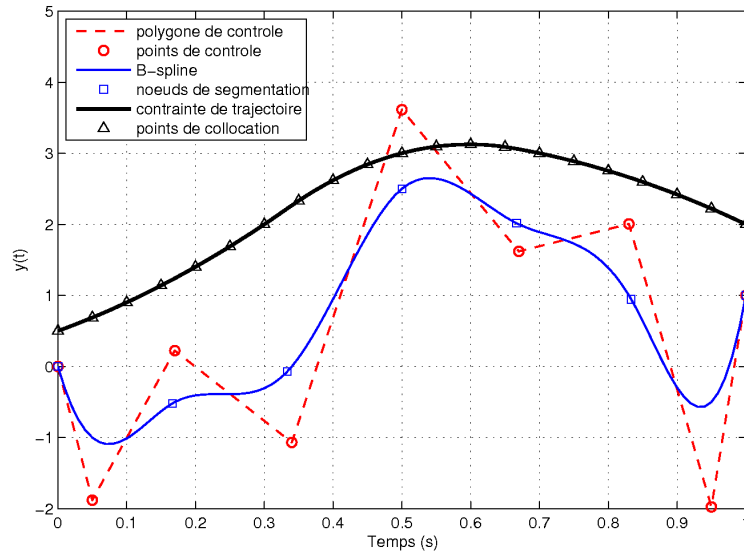


FIG. 2.20 – B-spline comportant 6 intervalles ( $l = 6$ ), d'ordre 4 ( $k = 4$ ), et de multiplicité 3 ( $s = 3$ ). Les contraintes sont satisfaites en 21 points de collocation régulièrement espacés. Les neuf points de contrôle représentent alors les variables de décision du problème.

**Remarque 2.18.** Bien entendu, il est également possible de distribuer les points de collocation de manière non uniforme, pas exemple en utilisant une répartition des points de type Gauss-Lobatto, ou Legendre-Gauss-Lobatto, exploitée notamment dans le cadre des techniques pseudo-spectrales d'optimisation de trajectoires [160].

Afin d'illustrer la démarche, considérons le problème de commande optimale (2.6). En paramétrant les variables d'optimisation (les sorties plates) par des B-splines et en les évaluant aux  $N_\tau$  points de collocation ( $\tau_1, \tau_2, \dots, \tau_{N_\tau}$ , la sous-matrice de collocation  $G_i$ , associée à la  $i^{\text{ème}}$  variable d'optimisation sera alors creuse [133], de la forme (2.61), où

$$Z_i(t) \triangleq \left( z_i^{(0)}(\tau_0), z_i^{(1)}(\tau_0), \dots, z_i^{(\beta_i)}(\tau_0), \dots, z_i^{(0)}(\tau_N), z_i^{(1)}(\tau_N), \dots, z_i^{(\beta_i)}(\tau_N) \right) \in \mathbb{R}^{(\beta_i+1) \times N},$$



que les points de collocation et les points de discrétisation du problème de commande optimale sont identiques. Nous les noterons simplement  $t_0 < t_1 < \dots < t_N$  dans la suite.

L'intégrale apparaissant dans la fonctionnelle de coût de l'OCP (2.6), définie par

$$\begin{aligned} J(u) &= \mathcal{C}_0(\psi_x(\bar{z}(t_0)), \psi_u(\bar{z}(t_0)), t_0) + \mathcal{C}_f(\psi_x(\bar{z}(t_f)), \psi_u(\bar{z}(t_f)), t_f) \\ &+ \int_{t_0}^{t_f} \mathcal{C}_t(\psi_x(\bar{z}(t)), \psi_u(\bar{z}(t)), t) dt, \end{aligned}$$

est approximée sous la forme d'une somme discrète en utilisant une *loi de quadrature*, telle que

$$\begin{aligned} J(u) &\approx \mathcal{C}_0(\psi_x(\bar{z}(t_0)), \psi_u(\bar{z}(t_0)), t_0) + \mathcal{C}_f(\psi_x(\bar{z}(t_f)), \psi_u(\bar{z}(t_f)), t_f) \\ &+ \sum_{j=0}^N \sum_{k=0}^P \mu_k \mathcal{C}_{t_k}(\psi_x(\bar{z}(t_{k+j})), \psi_u(\bar{z}(t_{k+j})), t_{k+j}), \end{aligned}$$

où les poids  $\mu_k$  et les points  $P$  sont imposés par la loi de quadrature retenue. De même, les contraintes linéaires et non linéaires de trajectoire associées au problème de commande optimale (2.6) s'écrivent

$$\begin{aligned} \Lambda_{lt}^j(u) &= A_t \bar{z}(t_j), & j &= 1, \dots, N, \\ \Lambda_{nlt}^j(u) &= c_t(\psi_x(\bar{z}(t_j)), \psi_u(\bar{z}(t_j))), & j &= 1, \dots, N. \end{aligned}$$

Les contraintes initiales et finales de l'OCP restent inchangées, et sont évaluées sous la forme

$$\begin{aligned} \Lambda_{li}(u) &= A_0 \bar{z}(t_0), \\ \Lambda_{lf}(u) &= A_f \bar{z}(t_f), \\ \Lambda_{nli}(u) &= c_0(\psi_x(\bar{z}(t_0)), \psi_u(\bar{z}(t_0))), \\ \Lambda_{nlf}(u) &= c_f(\psi_x(\bar{z}(t_f)), \psi_u(\bar{z}(t_f))). \end{aligned}$$

Notons  $\Lambda(u)$  l'ensemble des contraintes discrètes précédentes, tel que

$$\Lambda(u) = (\Lambda_{li}(u), \Lambda_{nli}(u), \Lambda_{lt}^1(u), \dots, \Lambda_{lt}^N(u), \Lambda_{nlt}^1(u), \dots, \Lambda_{nlt}^N(u), \Lambda_{lf}(u), \Lambda_{nlf}(u)).$$

La discrétisation précédente résulte en un problème de programmation non linéaire où les inconnues sont les points de contrôle  $u$  de l'ensemble des B-splines [133]. La structure générale du problème NLP est donnée par

$$\begin{aligned} \min_{u \in \mathbb{R}^M} \quad & J(u) & (2.62) \\ \text{t.q.} \quad & L_b \leq \Lambda(u) \leq U_b, \\ \text{où } M &= \sum_{i=1}^m l_i(k_i - s_i) + s_i. \end{aligned}$$

Finalement, le problème (2.62) peut être résolu en utilisant un solveur NLP efficace, tel que NPSOL [44], SNOPT [43], ou KNITRO [47].

## 2.7 Conclusion

Dans ce chapitre, nous avons décrit étape par étape une méthodologie originale permettant de planifier des trajectoires de rentrée atmosphérique en situation nominale, puis en présence d'un certain nombre de défauts actionneurs. La méthodologie proposée repose sur le concept de système plat et des techniques de transcription directe du problème de commande optimale.

Dans un premier temps, nous avons décrit l'architecture générale de la loi de guidage autonome proposée. Nous nous sommes ensuite focalisés sur l'utilisation de la propriété de platitude pour résoudre le problème de planification de trajectoires de rentrée atmosphérique en situation nominale. En partant d'un problème de commande optimale classique, défini dans l'espace d'état, nous avons vu qu'il était possible de réécrire ce dernier en fonction d'un nombre minimal de variables d'optimisation, qui sont alors constituées des sorties linéarisantes associées au système plat.

Dans un second temps, nous avons présenté une méthodologie générale permettant de convexifier le problème de commande optimale formulé dans l'espace des sorties plates. Le processus de convexification repose sur la résolution d'un problème d'annexion superellipsoïdale, dans lequel nous cherchons à approximer les contraintes de trajectoire et la fonctionnelle de coût du problème de commande optimale (supposées être non convexe) par des formes géométriques déformables, appelées superquadriques. Le problème d'optimisation associé est ensuite résolu grâce à un algorithme génétique afin d'obtenir une solution globale. Finalement, le problème de commande optimale décrit dans l'espace des sorties plates est reformulé de manière convexe, en utilisant les formes superquadriques obtenues.

Nous nous sommes ensuite attachés à développer une méthodologie permettant de prendre en compte l'effet d'un ou plusieurs défauts actionneurs dans le problème de commande optimale afin que la replanification de trajectoires soit tolérante aux pannes. L'idée consiste alors à intégrer dans le problème de commande optimale des contraintes liées à l'équilibre statique du véhicule ainsi qu'aux coefficients de forces aérodynamiques en situation défaillante. Deux solutions différentes sont proposées : la première tient compte de ces contraintes de manière explicite dans le problème de commande optimale formulé dans l'espace des sorties plates ; la seconde consiste à venir "piloter" le problème d'optimisation de trajectoires par un vecteur de paramètres exogène afin de replanifier des trajectoires à bord en présence d'une situation défaillante.

Enfin, la dernière section présente la technique de transcription directe du problème de commande optimal, formulé dans l'espace des sorties plates, en un problème de programmation non linéaire. Les variables d'optimisation sont alors paramétrées par une base particulière de polynômes continus par morceaux appelés B-splines. La dernière étape de la conversion repose alors sur une discrétisation du problème de commande optimale, afin d'obtenir un problème de programmation non linéaire qui sera résolu à bord du véhicule en utilisant un solveur adapté.

Dans le chapitre suivant, nous allons appliquer étape par étape l'ensemble de la démarche proposée à une mission de rentrée atmosphérique. Plus précisément, nous allons focaliser notre étude sur les phases de dissipation d'énergie en zone terminale et d'atterrissage automatique de la navette spatiale américaine STS-1.





## Chapitre 3

# Application à une mission de rentrée atmosphérique

“L’expérience ne se trompe jamais, ce sont nos jugements qui se trompent.”

Léonard de Vinci

---

### Résumé :

*Dans ce chapitre, nous appliquons étape par étape la méthodologie de guidage autonome décrite dans le chapitre 2 à une mission de rentrée atmosphérique, constituée des phases de dissipation d’énergie en zone terminale (TAEM), et d’approche et d’atterrissage automatique (A&L).*

*Le véhicule de rentrée considéré est la navette spatiale américaine STS-1, dont un simulateur complet a été développé dans le cadre de cette étude.*

*Après avoir décrit les objectifs et contraintes associés au guidage durant les phases TAEM et A&L, nous planifierons des trajectoires par platitude en considérant plusieurs formulations du problème de commande optimale. Nous décrirons également la méthode utilisée pour tenir compte de perturbations aérologiques, puis nous verrons comment convexifier le problème de commande optimale. Finalement, nous étudierons la replanification de trajectoires TAEM en présence d’une situation défaillante, permettant d’assurer que la trajectoire générée appartienne à l’enveloppe de vol réduite.*

---

## 3.1 Introduction

Ce chapitre est consacré à l'application de la méthodologie de guidage autonome proposée dans le chapitre précédent à une mission de rentrée atmosphérique constituée des phases de dissipation d'énergie en zone terminale (TAEM) et d'approche et atterrissage automatique (A&L).

Dans un premier temps, nous allons présenter la mission et le véhicule de rentrée atmosphérique considérés. Notamment, nous décrirons la géométrie des trajectoires TAEM et A&L, ainsi que les principaux objectifs et les différentes contraintes associés à la loi de guidage.

Nous utiliserons ensuite le concept de platitude différentielle afin de générer des trajectoires de rentrée à bord. Nous décrirons notamment la technique utilisée pour modéliser les coefficients de forces et de moments aérodynamiques, basée sur l'utilisation de décompositions en valeurs singulières et/ou de réseaux de neurones statiques.

Nous utiliserons ensuite la démarche d'annexion superellipsoïdale présentée dans le chapitre précédent afin de transformer le problème de commande optimale, formulé dans l'espace des sorties plates, en un problème d'optimisation convexe, dans lequel les contraintes de trajectoire ainsi que la fonctionnelle de coût sont approximés par des formes géométriques déformables appelées superquadriques.

Nous verrons ensuite qu'il est possible de tenir compte de certaines perturbations aérologiques stationnaires dans le problème de commande optimale associé à la planification de trajectoires à bord, en modélisant puis en intégrant des champs de cisaillement de vent tri-dimensionnels dans le problème de commande optimale.

Enfin, nous rendrons l'algorithme de re planification de trajectoires tolérant à certains types de défauts de gouvernes, en tenant compte de contraintes d'équilibrage statique dans le problème de commande optimale. Dans ce cadre, nous donnerons un certain nombre de résultats concernant la prise en compte explicite des contraintes d'équilibrage statique dans le problème de commande optimale, en présence d'une situation défaillante.

## 3.2 Problématique associée au guidage pour les phases TAEM et A&L

Nous débuterons cette section en rappelant les principales caractéristiques du véhicule de rentrée atmosphérique que nous allons considérer dans la suite de cette étude. Nous poserons ensuite la problématique associée à une mission de rentrée atmosphérique, plus précisément pour les phases de dissipation d'énergie en zone terminale, et d'approche finale et d'atterrissage automatique. De plus, nous donnerons pour chacune de ces phases la définition des modèles à trois degrés de liberté et des problèmes de commande optimale qui seront utilisés dans la section suivante pour planifier des trajectoires de rentrée

atmosphérique par platitude.

### 3.2.1 Navette spatiale américaine

L'US Space Shuttle, appelée officiellement Space Transportation System (STS), est le système de lancement spatial utilisé par le gouvernement américain pour effectuer des missions de vols habités. Au moment du lancement, le véhicule est constitué d'un réservoir principal externe (appelé external tank ou ET), de deux propulseurs d'appoint à poudre (appelés solid rocket boosters ou SRB) et de l'orbiteur (la navette), seul élément du lanceur à réellement aller dans l'espace (voir figure 3.1). Le programme Shuttle fut lancé peu après la fin du programme Apollo au début des années 70, à une époque où les restrictions budgétaires pour la NASA se faisaient déjà sentir. Cela conduisit à chercher les moyens les plus économiques pour accéder à l'espace et pour y maintenir une présence humaine. Chaque navette a été conçue pour effectuer une centaine de lancements. L'un des principaux objectifs du programme STS était alors de construire et de desservir une station spatiale orbitale, ce qui est en train de se réaliser avec l'assemblage final de la Station spatiale Internationale (ISS), et comme ce fut le cas à l'époque avec la station MIR.

Lors du décollage, la navette est propulsée par les deux propulseurs d'appoint à poudre SRB qui sont placés de part et d'autre du réservoir externe. La navette possède également trois types de moteurs différents : les moteurs principaux SSME (Space Shuttle Main Engines), les propulseurs OMS (Orbital Maneuvering System), et le système de contrôle d'attitude RCS (Reaction Control System). Les SSME, au nombre de trois (voir figure 3.2), constituent les propulseurs principaux du lanceur après que les SRB aient consommé tout le combustible solide dont ils sont composés, et qu'il aient été largués<sup>49</sup> à une altitude d'environ 45 km. Les moteurs principaux SSME consomment le contenu de l'External Tank (ET) en huit minutes environ. Lorsque la navette a brûlé tout le contenu du réservoir externe, elle le largue à environ 113 km. Ce dernier rentre alors dans l'atmosphère et se désintègre. Pendant ce temps, le système de manoeuvres orbitales OMS (Orbital Manoeuvring System) fournit à la navette la dernière poussée qui lui permet d'atteindre sa vitesse de satellisation. Les tuyères du système OMS sont situées de part et d'autre du stabilisateur vertical. Elles sont allumées une dernière fois avant la rentrée dans l'atmosphère, lors de la phase de désorbitation de la navette. Cette dernière effectue alors une manoeuvre de retournement afin de se positionner dans le sens inverse du vol, et les OMS sont mis à feu de manière à la freiner. Le système de contrôle d'attitude, dit RCS (Reaction control System), est alors utilisé afin de retourner la navette pour qu'elle présente une attitude correcte au point d'injection<sup>50</sup>, puis entame son retour sur Terre. Les tuyères du système RCS, placées à l'arrière mais aussi à l'avant et sur les côtés du véhicule (voir figure 3.2), permettent de petits allumages brefs et à faible puissance.

Au début de la rentrée, dans les hautes couches de l'atmosphère, le système de contrôle

---

<sup>49</sup>les SRB sont alors récupérés dans l'océan à des fins d'expertise.

<sup>50</sup>roulis et lacet nuls, et incidence prédéfinie.

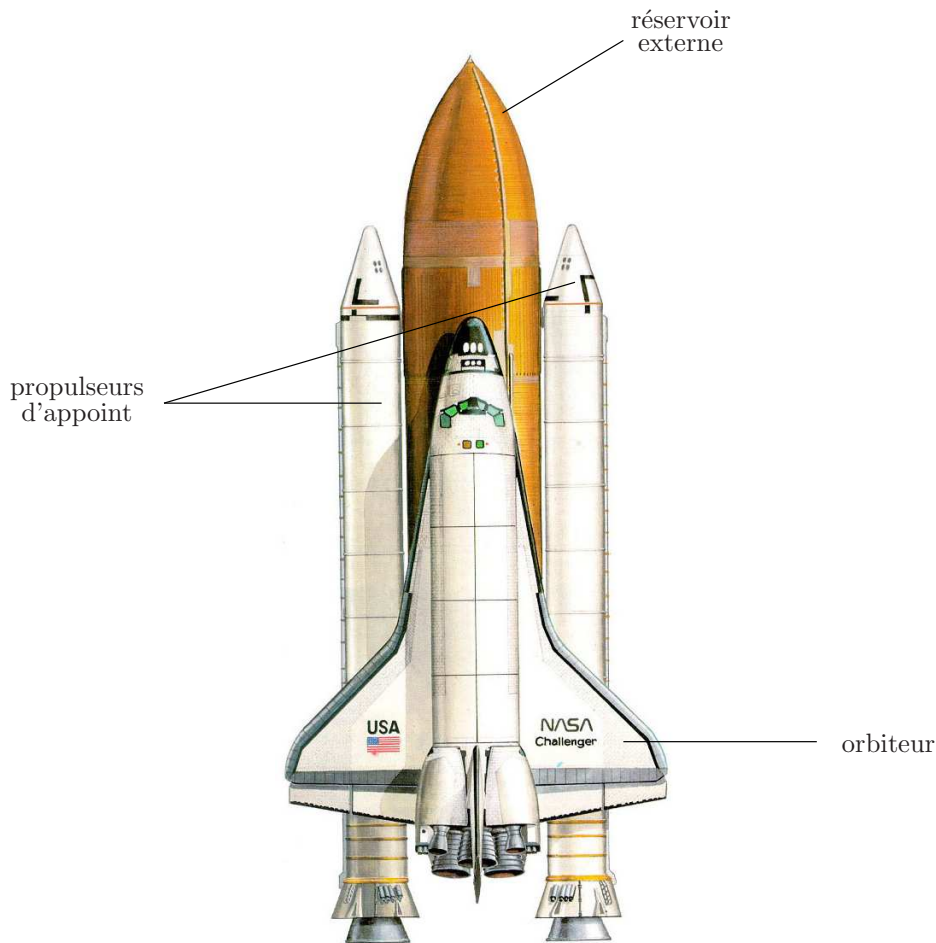


FIG. 3.1 – illustration du système de lancement Shuttle (STS).

d'attitude RCS est principalement utilisé pour manoeuvrer la navette, puis les gouvernes prennent ensuite le relais lorsque la pression dynamique devient supérieure à 480 Pa. Les différentes gouvernes équipant le véhicule sont les élevons situés sur chaque aile, la gouverne de direction qui joue également le rôle d'aérofrein en s'ouvrant de manière symétrique, et le body flap situé en dessous des propulseurs principaux de la navette. Le body flap permet d'assurer l'équilibrage longitudinal de l'orbiteur lors de la rentrée atmosphérique. Les quatre élevons permettent de contrôler à la fois le roulis et le tangage de l'appareil en fonction de leur braquage symétrique ou différentiel. Le tableau 3.1, extrait de la base de données aérodynamique de la navette STS-1 [166], présente les limites et vitesses de braquage associés à chaque surface de contrôle aérodynamique. Selon leurs valeurs de braquage, les quatre élevons peuvent jouer le rôle de gouverne de profondeur ou d'ailerons. En particulier, la valeur moyenne des braquages des élevons fournit le braquage  $\delta_e$  agissant sur la profondeur, tel que

$$\delta_e = \frac{1}{2}(\delta_{e_i} + \delta_{e_o}) = \frac{1}{2}(\delta_{e_l} + \delta_{e_r}), \quad (3.1)$$

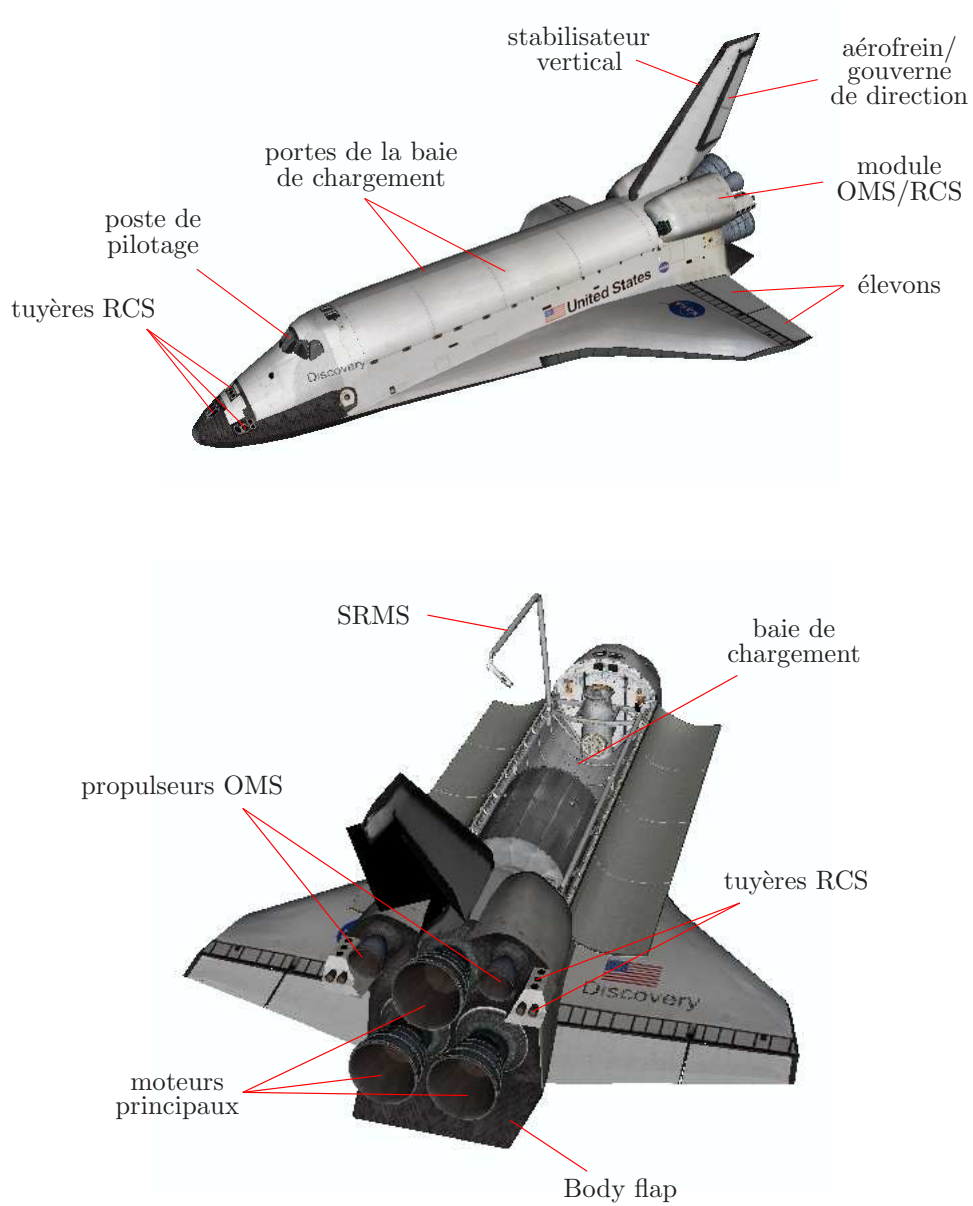


FIG. 3.2 – description de l'orbiteur.

avec

$$\begin{aligned} \delta_{e_i} &= \frac{1}{2}(\delta_{e_{i_l}} + \delta_{e_{i_r}}), \\ \delta_{e_o} &= \frac{1}{2}(\delta_{e_{o_l}} + \delta_{e_{o_r}}), \\ \delta_{e_l} &= \frac{1}{2}(\delta_{e_{i_l}} + \delta_{e_{o_l}}) = \delta_e + \delta_a, \\ \delta_{e_r} &= \frac{1}{2}(\delta_{e_{i_r}} + \delta_{e_{o_r}}) = \delta_e - \delta_a. \end{aligned}$$

gouverne	symbole	limites de braquage		vitesses de braquage (deg/s)
		Min (deg)	Max (deg)	
<b>élevons</b>				
profondeur	$\delta_e$	-35	20	20
ailerons	$\delta_a$	-35	20	20
<b>gouverne de direction</b>	$\delta_r$	-22.8	22.8	10
<b>aérofrein</b>	$\delta_{sb}$	0	87.2	5
<b>body flap</b>	$\delta_{bf}$	-11.7	22.55	1.3

TAB. 3.1 – limites et vitesses de braquage des gouvernes.

De plus, les braquages différentiels des élevons fournissent un braquage  $\delta_a$ , agissant sur le gauchissement de l'appareil, tel que

$$\delta_a = \frac{1}{2}(\delta_{e_l} - \delta_{e_r}) = \frac{1}{2}(\delta_{a_i} + \delta_{a_o}), \quad (3.2)$$

où les termes  $\delta_{a_i}$  et  $\delta_{a_o}$  sont définis respectivement par

$$\begin{aligned} \delta_{a_i} &= \frac{1}{2}(\delta_{e_{i_l}} - \delta_{e_{i_r}}), \\ \delta_{a_o} &= \frac{1}{2}(\delta_{e_{o_l}} - \delta_{e_{o_r}}). \end{aligned}$$

**Remarque 3.1.** La gouverne de direction est constituée de deux parties mobiles indépendantes qui jouent à la fois le rôle de gouverne de direction et d'aérofrein.

La NASA a construit en tout sept navettes, et 126 vols<sup>51</sup> de celles-ci ont été effectués au 11 mai 2009.

- (1976) Enterprise, navette expérimentale, qui n'a jamais été mise en orbite.
- (1977) Pathfinder, maquette conçue pour effectuer des essais au sol.
- (1981) Columbia, désintégrée lors de son retour dans l'atmosphère le 1<sup>er</sup> février 2003.
- (1982) Challenger, détruite lors du décollage le 28 janvier 1986.
- (1983) Discovery.
- (1985) Atlantis.
- (1991) Endeavour, construite afin de remplacer Challenger.

Nous avons considéré dans le cadre de cette étude la navette spatiale américaine STS-1, qui effectua le tout premier vol spatial du programme Shuttle. L'orbiteur en question est Columbia ou OV-102 (Orbital Vehicle 102), qui décolla le 12 avril 1981 du centre spatial Kennedy en Floride.

Le tableau suivant regroupe les caractéristiques principale de l'orbiteur STS-1. Le programme US Space Shuttle a été entaché de deux accidents qui ont entraîné la perte des équipages :

<sup>51</sup>133 vols sont prévus en tout jusqu'à la mise à la retraite des navettes en 2010.

caractéristiques	symboles	valeurs
surface de référence [ $m^2$ ]	$S$	249.9
masse totale au point d'injection [ $kg$ ]	$m$	89930
envergure des ailes [ $m$ ]	$b$	23.8
corde de référence [ $m$ ]	$\bar{c}$	12
finesse maximale (pour $M \leq 3$ )	$(L/D)_{\max}$	$\approx 4$
	$I_{xx}$	1213866
moments d'inertie [ $kg/m^2$ ]	$I_{yy}$	9378654
	$I_{zz}$	9759518
	$I_{xz}$	228209
produits d'inertie [ $kg/m^2$ ]	$I_{xy}$	6136
	$I_{yz}$	2972
	$x_{mrc}$	17
centre de référence des moments [ $m$ ]	$y_{mrc}$	0
	$z_{mrc}$	-1.2
	$x_{cg}$	27.3
centre de gravité [ $m$ ]	$y_{cg}$	0
	$z_{cg}$	9.5

TAB. 3.2 – caractéristiques principales de la navette spatiale américaine STS-1.

- la destruction au décollage de la navette Challenger, le 28 janvier 1986, suite à la rupture d'un joint sur un propulseur d'appoint SRB.
- la désintégration dans l'atmosphère de la navette Columbia le 1er février 2003, suite à la détérioration de son bouclier thermique lors du décollage.

Ces deux accidents ont globalement mis en cause la mauvaise gestion du programme Shuttle par la NASA, notamment le fait qu'elle ne prenait pas assez en considération la sécurité des vols alors que la navette spatiale demeure, encore aujourd'hui, un engin expérimental. L'annonce en 2004 par le Président des États-Unis George W. Bush de relancer l'exploration habitée de la Lune et de préparer le voyage interplanétaire vers Mars marque la fin du programme Shuttle, ce dernier ne pouvant y jouer aucun rôle majeur. Les trois orbiteurs restants seront donc mis à la retraite avant d'être probablement exposés dans un musée à l'instar du démonstrateur Enterprise d'ici 2015.

### 3.2.2 Phases de dissipation d'énergie en zone terminale et d'atterrissage automatique

Une mission typique de rentrée atmosphérique, pour un véhicule dit "planant"<sup>52</sup>, est divisée en trois phases distinctes car les objectifs et contraintes associés à chacune de ces phases leur sont propres. La figure 3.3 illustre une mission typique de rentrée atmosphérique, depuis le point d'injection<sup>53</sup> sur la trajectoire hypersonique jusqu'au toucher des

<sup>52</sup>classiquement, un véhicule est dit "planant" si sa finesse est supérieure ou égale à 2 environ.

<sup>53</sup>après avoir effectué une manoeuvre de désorbitation.



roues sur la piste. Les trois phases représentées sur la figure 3.3 sont donc :

1. la *phase hypersonique*, qui débute au point d'injection à environ 120 km d'altitude, et se termine au passage de Mach 2. L'objectif principal de cette phase est de limiter le flux agissant sur le bouclier thermique de l'engin, tout en dissipant son énergie en effectuant des manoeuvres de *renverses de roulis*<sup>54</sup>. Le facteur de charge et la pression dynamique sont également contrôlés afin de ne pas dépasser des valeurs prédéfinies, liées à la conception du véhicule.
2. la *phase de dissipation d'énergie en zone terminale* (TAEM) qui, comme son nom l'indique, est principalement dédiée à dissiper l'énergie du véhicule. Ce segment de vol débute au point d'entrée TAEM (TEP) et se termine à l'interface avec la phase d'atterrissage automatique (ALI), définie lorsque le véhicule est aligné avec l'axe de la piste et satisfait des conditions cinématiques données. Le flux thermique agissant de manière très faible sur le véhicule lors de cette phase, les seules contraintes mécaniques considérées sont le facteur de charge total et la pression dynamique.
3. la *phase d'approche finale et d'atterrissage automatique* (A&L), qui débute au point ALI, et se termine à l'arrêt des roues sur la piste. Cette phase est effectuée à la manière d'un avion de transport civil, à la différence près que la pente initiale sur la trajectoire de descente est relativement forte, due à l'absence de moyens propulsifs et à la finesse relativement faible du véhicule.

La phase TAEM est souvent décrite par un ensemble de segments géométriques adjacents. La géométrie de nous avons choisi de décrire ci-après est basée sur la stratégie adoptée pour la navette spatiale Russe Buran [173]. La principale différence avec l'approche utilisée pour la navette spatiale américaine, décrite en détails par T. E. Moore dans [85], est qu'un *cylindre d'alignement* (fictif), appelé HAC, est utilisé au lieu d'un cône d'alignement (également appelé HAC), afin d'aligner le véhicule avec l'axe de piste. De plus, les segments géométriques décrivant la phase TAEM sont réalisés dans un ordre différent de celui utilisé pour l'approche américaine (cf [85]). La figure 3.4 illustre la stratégie utilisée pour le guidage de la navette Buran.

Typiquement, le calculateur de bord effectue les fonctions de guidage du véhicule dans l'ordre suivant :

1. *virages de dissipation en S* : si le véhicule possède un niveau d'énergie trop important, ces virages préliminaires permettent de dissiper l'excédent d'énergie en écartant le véhicule du HAC jusqu'à ce qu'une valeur d'énergie prédéfinie soit atteinte.
2. *phase d'acquisition du HAC* : cette phase est constituée d'un virage permettant d'orienter le nez du véhicule vers un point tangent au HAC.
3. *phase d'interception du HAC* : le véhicule vole alors à ailes droites jusqu'à ce que le point tangent au HAC soit atteint.
4. *phase d'alignement de cap* : le véhicule est asservi en position autour du HAC, depuis le point de tangence jusqu'au point ALI, au niveau duquel le nez du véhicule est aligné avec l'axe de piste.

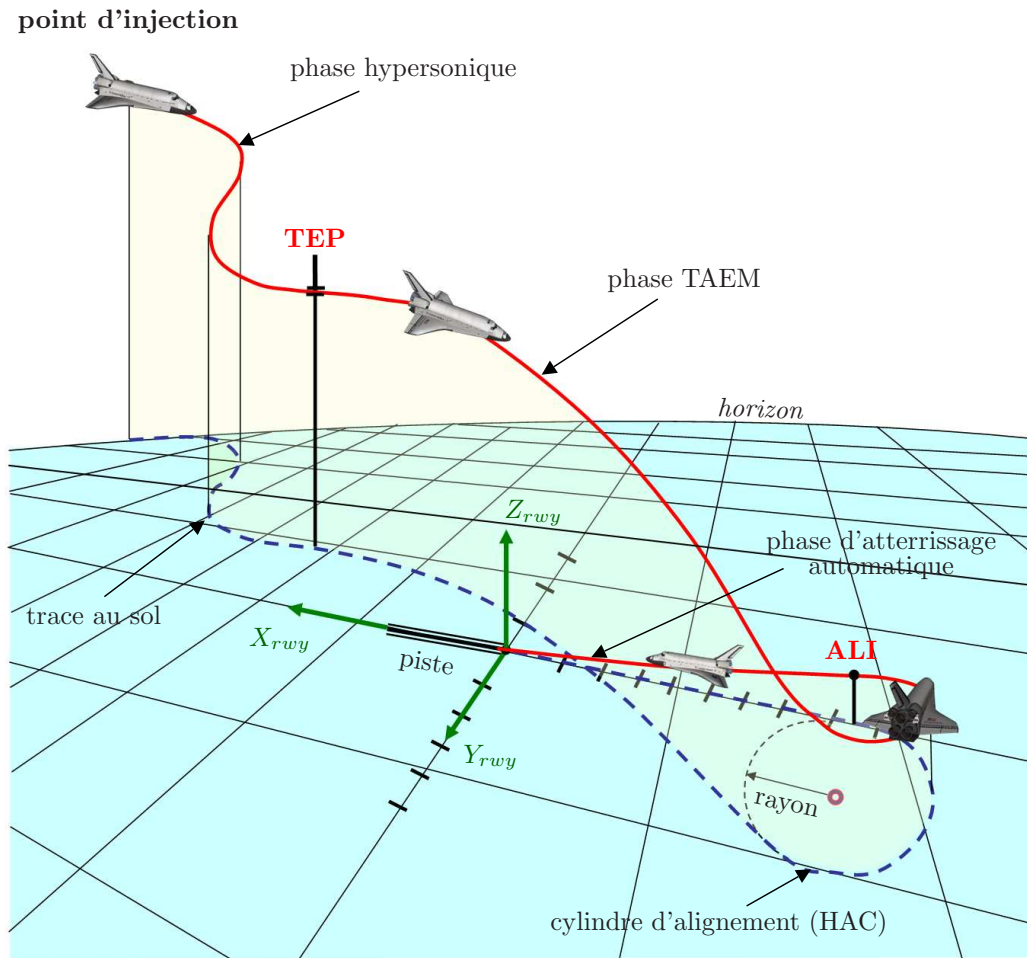


FIG. 3.3 – illustration d’une mission de rentrée atmosphérique.

**Remarque 3.2.** La stratégie de guidage TAEM utilisée pour la navette spatiale américaine comprend une phase de vol supplémentaire, appelée *pré-finale*, débutant lorsque le véhicule est aligné avec l’axe de piste, et se terminant au point ALI, après que le véhicule est effectué une portion de trajectoire de descente équilibrée à ailes horizontales.

**Remarque 3.3.** La stratégie de guidage proposée n’étant pas fondée sur un partitionnement de la trajectoire de vol en plusieurs segments, mais utilisant plutôt une formulation optimale du problème de planification de trajectoires, la description géométrique précédente permet uniquement au lecteur de mieux appréhender le problème traité dans ce chapitre.

Les objectifs associés au développement d’une loi de guidage pour la phase TAEM concernent surtout la planification de trajectoires, le régulateur de poursuite de trajectoire agissant uniquement de manière locale autour de la trajectoire de vol courante. Ces objectifs sont doubles :

<sup>54</sup>à la manière d’un skieur alpin qui descend une piste en effectuant des virages en forme de S.

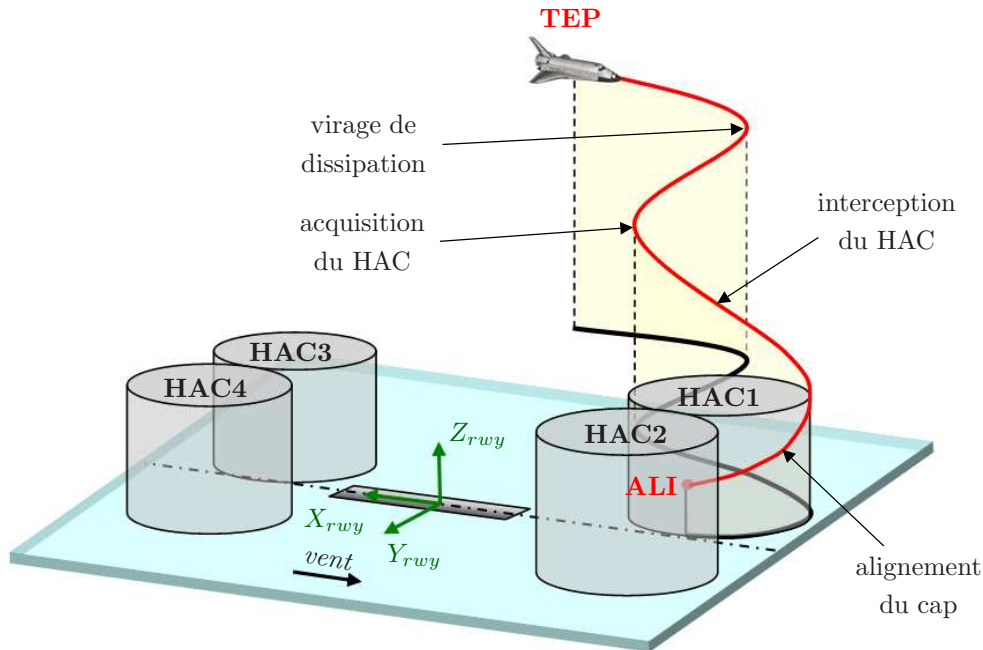


FIG. 3.4 – phase de dissipation d'énergie en zone terminale (TAEM).

- Premièrement, une protection doit être mise en place vis-à-vis des contraintes mécaniques excessives susceptibles d'agir sur le véhicule (facteur de charge total et pression dynamique, la vitesse de l'engin étant trop faible pour prendre en considération le flux thermique).
- Deuxièmement, les contraintes de trajectoires liées aux conditions cinématiques finales au point ALI (vitesse relative, pente sur la trajectoire, alignement avec l'axe de piste) doivent être satisfaites afin de permettre un atterrissage fiable de l'engin<sup>55</sup>. L'ensemble des objectifs de mission et des contraintes mécaniques sont résumés dans le tableau 3.3 dans le cas de la navette spatiale américaine.

De plus, des contraintes additionnelles, liées aux ressources énergétiques du véhicule en situation nominale, sont associées aux angles d'incidence et de gîte, ainsi que sur leurs vitesses angulaires, tel qu'indiqué dans le tableau 3.4. Ces contraintes sont implicitement liées à celles associées aux surfaces de contrôle aérodynamique définies dans le tableau 3.1.

L'objectif principal de la loi de guidage est alors de générer les consignes de guidage en boucle fermée de telle manière que le véhicule puisse atteindre des conditions cinématiques prédéfinies au point ALI, tout en s'assurant que les différentes contraintes de trajectoire soient respectées, et en tenant compte de conditions énergétiques variées au point TEP. Les dispersions cinématiques initiales au point TEP proviennent des incertitudes et imprécisions finales engendrées par la loi de guidage associée à la phase de rentrée hypersonique.

<sup>55</sup>parfois, le survol d'une région habitée doit également être évité pour des raisons évidentes de sécurité.

	objectifs
<b>Contraintes mécaniques de trajectoire</b>	
facteur de charge maximal $\Gamma_{\max}$ [ $g$ ]	< 2.5
pression dynamique maximale $\bar{q}_{\max}$ [ $kPa$ ]	< 16
<b>Contraintes cinématiques au point ALI</b>	
nombre de Mach	0.5
altitude [ $km$ ]	5
portée longitudinale [ $km$ ]	10
portée latérale [ $km$ ]	0
cap final [ $deg$ ]	vent de face
pente sur la trajectoire [ $deg$ ]	-27

TAB. 3.3 – contraintes de guidage associées à la phase TAEM.

	$\alpha$	$\mu$	$\beta$
valeur minimale [ $deg$ ]	0	-80	-3
valeur maximale [ $deg$ ]	25	80	3
vitesse angulaire maximale [ $deg/s$ ]	2	5	2

TAB. 3.4 – contraintes associées aux consignes de guidage.

Nous sommes maintenant en mesure de décrire le modèle à trois degrés de liberté régissant le mouvement du centre de gravité du véhicule dans un repère cartésien local ( $X_{rwy}, Y_{rwy}, Z_{rwy}$ ) attaché au centre de la piste (voir figure 3.4). Sous l’hypothèse d’une Terre non tournante, qui est compatible avec le dimensionnement de la trajectoire TAEM (durée totale relativement faible), le véhicule peut être positionné par rapport à un repère cartésien local lié au sol, souvent attaché à l’axe de piste, et associé à la projection conique conforme de Lambert. De ce fait, les effets de la force de Coriolis et de la force centrifuge sur la dynamique du véhicule ne sont pas pris en compte. Les coordonnées relatives du véhicule par rapport à ce repère sont alors communément appelées coordonnées “*Terre plate*”, faisant référence au système de projection utilisé.

Sous les hypothèses précédentes, les équations régissant la dynamique du centre de gravité du véhicule de rentrée par rapport au repère sol local ( $X_{rwy}, Y_{rwy}, Z_{rwy}$ ) sont alors définies

par [32] (voir également l'annexe C)

$$\dot{x} = V \cos \chi \cos \gamma, \quad (3.3)$$

$$\dot{y} = V \sin \chi \cos \gamma, \quad (3.4)$$

$$\dot{h} = V \sin \gamma, \quad (3.5)$$

$$\dot{V} = -\frac{D(\alpha, M)}{m} - g \sin \gamma, \quad (3.6)$$

$$\dot{\gamma} = \frac{1}{mV} (L(\alpha, M) \cos \mu - Y(\beta, M) \sin \mu) - \frac{g}{V} \cos \gamma, \quad (3.7)$$

$$\dot{\chi} = \frac{1}{mV \cos \gamma} (L(\alpha, M) \sin \mu + Y(\beta, M) \cos \mu), \quad (3.8)$$

où  $(x, y, h, V, \gamma, \chi)$  est l'état, comprenant respectivement la portée longitudinale  $x$ , la portée latérale  $y$ , l'altitude  $h$ , la vitesse  $V$ , la pente sur la trajectoire  $\gamma$ , et le cap  $\chi$ . Les entrées de commande du système sont l'incidence  $\alpha$ , le dérapage  $\beta$  et l'angle de gîte  $\mu$ . La force de portance  $L$ , de traînée  $D$  et la force latérale  $Y$  dépendent des coefficients aérodynamiques en configuration lisse, telles que

$$L(\alpha, M) = \bar{q} S C_{L_0}(\alpha, M), \quad (3.9)$$

$$D(\alpha, M) = \bar{q} S C_{D_0}(\alpha, M). \quad (3.10)$$

$$Y(\beta, M) = \bar{q} S C_{Y_0}(\beta, M), \quad (3.11)$$

où

$$\bar{q} = \frac{1}{2} \rho V^2 \quad (3.12)$$

est la pression dynamique, et  $C_{L_0}(\alpha, M)$ ,  $C_{D_0}(\alpha, M)$  et  $C_{Y_0}(\beta, M)$  représentent respectivement les coefficients de portance, de traînée et de force latérale en configuration lisse. Nous verrons dans la section (3.3) la technique utilisée pour modéliser les coefficients de forces et de moments aérodynamiques par des relations analytiques dépendant d'un nombre réduit de paramètres.

Nous supposons en outre que la masse  $m$  du véhicule est constante durant toute la phase TAEM (rentrée non propulsée), et que l'accélération gravitationnelle  $g$  est constante. De plus, un modèle exponentiel défini par

$$\rho = \rho_0 \exp(-h/H_0), \quad (3.13)$$

est utilisé afin de décrire le profil de la densité atmosphérique en fonction de l'altitude, dans lequel  $\rho_0$  représente la densité de l'air au niveau moyen de la mer et  $H_0$  est un facteur d'échelle atmosphérique constant ajusté aux couches basses de l'atmosphère.

**Remarque 3.4.** Notons qu'il est nécessaire de définir un modèle analytique différentiable de la densité atmosphérique afin de pouvoir exploiter la propriété de platitude du modèle (3.3)-(3.8). Nous aurions pu également modéliser la variation de la densité atmosphérique en fonction de l'altitude par une relation polynomiale, ajustée grâce à une table d'atmosphère standard du type US 1969, US 1976, ou encore CIRA 88.

La contrainte mécanique associée au facteur de charge total peut s'écrire sous la forme

$$\Gamma = \frac{\sqrt{L^2(\alpha, M) + D^2(\alpha, M)}}{mg}. \quad (3.14)$$

**Remarque 3.5.** La contrainte de flux thermique au point d'arrêt, qui constitue l'une des contraintes les plus importantes du guidage hypersonique (et qui n'est pas considérée pour les phases TAEM et A&L), est typiquement définie par

$$\Phi = C_q \sqrt{\rho} V^3,$$

où  $C_q$  est une *constante de flux* qui dépend des caractéristiques aérothermodynamiques du véhicule de rentrée considéré.

Finalement, l'ensemble des objectifs et des contraintes associés à la phase TAEM donne naissance au problème de commande optimale suivant :

$$\begin{aligned} \min_{x(t), u(t)} \quad & \mathcal{C}_0(x(t_0), u(t_0)) + \int_{t_0}^{t_f} \mathcal{C}_t(x(t), u(t)) dt + \mathcal{C}_f(x(t_f), u(t_f)) \quad (3.15) \\ \text{t.q.} \quad & \dot{x}(t) = f(x(t), u(t)), \quad t \in [t_0, t_f], \\ & x(t_0) = x_0, \\ & u(t_0) = u_0, \\ & 0 \leq \Gamma(x(t), u(t)) \leq \Gamma_{\max}, \quad t \in [t_0, t_f], \\ & 0 \leq \bar{q}(x(t)) \leq \bar{q}_{\max}, \quad t \in [t_0, t_f], \\ & u_{\min} \leq u(t) \leq u_{\max}, \quad t \in [t_0, t_f], \\ & x(t_f) = x_f, \\ & u(t_f) = u_f, \end{aligned}$$

où l'état  $x$  est défini par  $(x, y, h, V, \gamma, \chi)$  et le vecteur de commandes par  $u = (\alpha, \beta, \mu)$ . La fonctionnelle de coût utilisée dépend des objectifs de mission considérés.

**Remarque 3.6.** Les contraintes finales sur les commandes  $u(t_f) = u_f$  introduites dans le problème de commande optimale (3.15) peuvent être omises dans certains cas. En revanche, il peut être intéressant de spécifier par exemple un vol à ailes droites lors du passage au point ALI, i.e., avec  $\mu = 0$ , ou encore à dérapage nul  $\beta = 0$ , afin d'amorcer la phase d'atterrissage automatique avec de "bonnes" conditions initiales.

Intéressons-nous maintenant de plus près à la phase d'atterrissage automatique (A&L) de la navette spatiale américaine, qui est en plusieurs points similaires à celle du X-34 décrite dans la section 1.5.1. La phase d'atterrissage automatique est une phase de vol située dans un plan vertical qui ne fait intervenir que la dynamique longitudinale du véhicule. Elle débute au point d'entrée ALI (voir figure 3.5), lorsque le nombre de Mach est égal à 0.5 (voir les conditions TAEM finales dans le tableau 3.3), et se termine lorsque le véhicule s'arrête sur la piste. De même que pour la trajectoire d'atterrissage automatique du X-34, la trajectoire comprend trois principaux segments :

1. une phase de descente équilibrée parcourue avec une pente forte constante  $\gamma_1$ , permettant d'annuler les dispersions sur les conditions cinématiques initiales,

2. un arrondi circulaire réalisé à facteur de charge  $g$  constant, permettant de faire croître la pente de descente,
3. une phase de descente équilibrée parcourue avec une pente faible constante  $\gamma_2$  (et avec une vitesse verticale relativement faible), jusqu'à ce que l'arrondi final soit déclenché peu avant le toucher des roues.

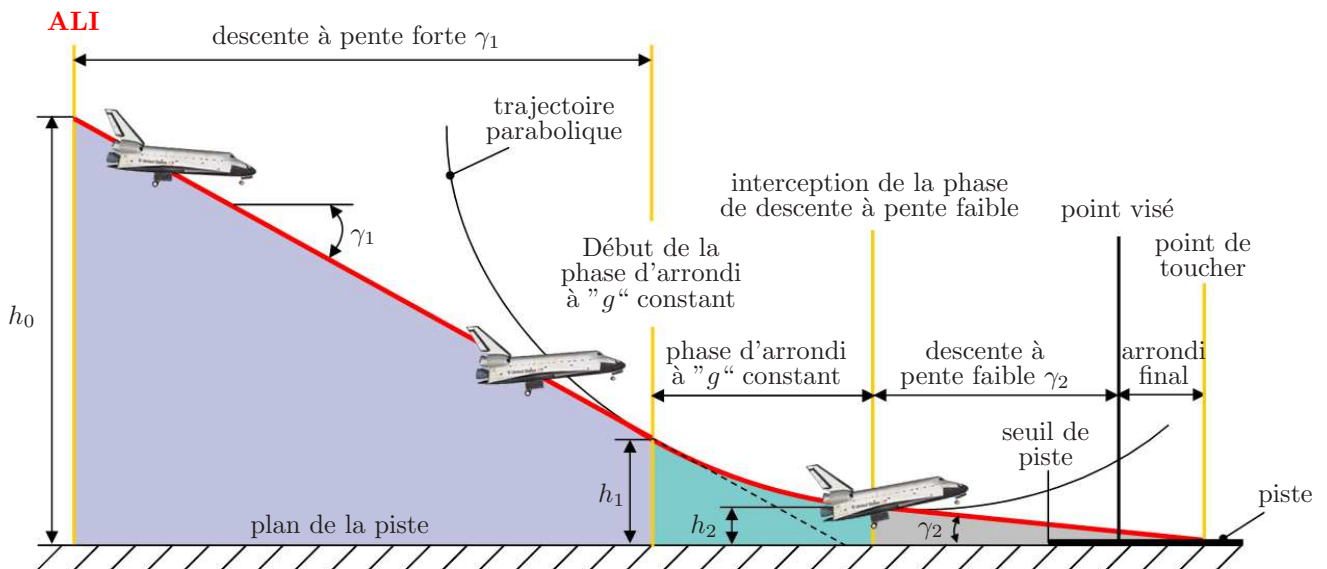


FIG. 3.5 – phase d'atterrissage automatique (A&L).

**Remarque 3.7.** Dans le cas présent, il n'y a pas de phase de décroissance exponentielle permettant de supprimer la discontinuité d'accélération existant entre l'arrondi effectué à facteur de charge constant et la phase de descente à pente faible.

**Remarque 3.8.** Le problème de planification de trajectoires d'atterrissage automatique étant formulé comme un problème de commande optimale, nous ne considérerons pas dans la suite le partitionnement géométrique décrit précédemment.

Les contraintes mécaniques associées à la phase A&L sont similaires à celles rencontrées lors de la phase TAEM (facteur de charge et pression dynamique). De même, les bornes inférieures et supérieures associées aux angles d'incidence, de gîte, de dérapage, ainsi que leurs vitesses angulaires maximales, demeurent ici inchangés (voir tableau 3.4). Les contraintes cinématiques associées à la phase d'atterrissage automatique sont imposés au point de toucher des roues sur la piste. Notamment, il est impératif que la pente finale sur la trajectoire (ou, de manière équivalente, la vitesse de descente verticale) ainsi que la vitesse relative (où, de manière équivalente, la pression dynamique) n'excèdent pas des valeurs prédéfinies afin que le véhicule ne subisse pas de dommages matériels. Les contraintes de trajectoire ainsi que les contraintes cinématiques finales associées à la phase A&L sont résumées dans le tableau 3.5 dans le cas de la navette spatiale américaine.



	objectifs
<b>Contraintes mécaniques de trajectoire</b>	
facteur de charge maximal $\Gamma_{\max}$ [g]	< 2.5
pression dynamique maximale $\bar{q}_{\max}$ [kPa]	< 16
<b>Contraintes cinématiques au point de toucher</b>	
vitesse relative [m/s]	90
altitude [km]	altitude de la piste
portée longitudinale [km]	0
pente sur la trajectoire [deg]	-3

TAB. 3.5 – contraintes de guidage associées à la phase A&L.

Le modèle à trois degrés de liberté associé à la phase d’atterrissage automatique peut également être décrit en utilisant l’hypothèse “Terre plate”. Les équations dynamiques longitudinales régissant le mouvement du centre de gravité du véhicule par rapport au repère cartésien local  $(X_{rwy}, Y_{rwy}, Z_{rwy})$  lié à la piste, sont données par [32]

$$\dot{x} = V \cos \gamma, \quad (3.16)$$

$$\dot{h} = V \sin \gamma, \quad (3.17)$$

$$\dot{V} = -\frac{D(\alpha, M)}{m} - g \sin \gamma, \quad (3.18)$$

$$\dot{\gamma} = \frac{L(\alpha, M)}{mV} - \frac{g}{V} \cos \gamma, \quad (3.19)$$

où  $x$  représente la portée longitudinale,  $h$  l’altitude,  $V$  la vitesse relative du véhicule,  $\gamma$  sa pente. L’unique commande présente est l’angle d’incidence  $\alpha$ . De plus, les forces de portance  $L$  et de traînée  $D$  sont définies respectivement par (3.9) et (3.10).

**Remarque 3.9.** Clairement, les équations (3.16)-(3.19) constituent un cas particulier des équations (3.3)-(3.8) correspondant à la phase de dissipation d’énergie en zone terminale, obtenues en conservant uniquement les expressions associées au mouvement longitudinal du véhicule.

Les contraintes mécaniques de facteur de charge et de pression dynamique sont calculées grâce aux équations (3.14) et (3.12) respectivement.

Comme pour la phase TAEM, l’ensemble des contraintes associées au problème de planification de trajectoires A&L peut être intégré dans un problème de commande optimale,



qui s'écrit de manière similaire sous la forme

$$\begin{aligned}
 \min_{x(t), u(t)} \quad & \mathcal{C}_0(x(t_0), u(t_0)) + \int_{t_0}^{t_f} \mathcal{C}_t(x(t), u(t)) dt + \mathcal{C}_f(x(t_f), u(t_f)) \quad (3.20) \\
 \text{t.q.} \quad & \dot{x}(t) = f(x(t), u(t)), \quad t \in [t_0, t_f], \\
 & x(t_0) = x_0, \\
 & u(t_0) = u_0, \\
 & 0 \leq \Gamma(x(t), u(t)) \leq \Gamma_{\max}, \quad t \in [t_0, t_f], \\
 & 0 \leq \bar{q}(x(t)) \leq \bar{q}_{\max}, \quad t \in [t_0, t_f], \\
 u_{\min} \leq & u(t) \leq u_{\max}, \quad t \in [t_0, t_f], \\
 & x(t_f) = x_f, \\
 & u(t_f) = u_f,
 \end{aligned}$$

où l'état  $x$  est défini par  $(x, h, V, \gamma)$  et la commande est l'angle d'incidence  $u = \alpha$ . La fonctionnelle de coût apparaissant dans le problème (3.20) dépend des objectifs de mission considérés pour la phase d'atterrissage automatique.

Dans la section suivante, nous allons résoudre les problèmes de commande optimale (3.15) et (3.20), associés aux phases TAEM et A&L, en utilisant la propriété de platitude respective des modèles (3.3)-(3.8) et (3.16)-(3.16).

### 3.3 Planification de trajectoires de rentrée atmosphérique par platitude

Nous avons montré dans la proposition 3.1 du dernier chapitre de la première partie de ce mémoire que le modèle à trois degrés de liberté d'un véhicule de rentrée atmosphérique (3.3)-(3.8) est plat, et que la sortie plate est constituée de la position cartésienne du véhicule dans le repère local  $(X_{rwy}, Y_{rwy}, Z_{rwy})$  lié au sol. Nous allons dans cette section utiliser cette propriété de platitude afin de transcrire dans l'espace des sorties plates les problèmes de commande optimale (3.15) et (3.20) associés aux phases de dissipation d'énergie en zone terminale (TAEM), et d'approche et d'atterrissage automatique (A&L). Mais avant cela, il est nécessaire de modéliser les coefficients de forces et de moments aérodynamiques par des expressions analytiques différentiables. C'est l'objet de la section suivante.

#### 3.3.1 Modélisation des coefficients aérodynamiques

Comme indiqué par J. T. Betts dans son ouvrage sur l'optimisation de trajectoires [34], l'approximation de tables de données discrètes par des fonctions analytiques différentiables revêt une importance capitale, car les algorithmes numériques de résolution sont souvent basés sur une hypothèse de différentiabilité des fonctions considérées, notamment pour le calcul du Jacobien ou du Hessien. De nombreuses méthodes peuvent être utilisées à cette fin. Pour certaines applications, des fonctions polynomiales suffiront, alors que

pour d'autres, une modélisation plus fine sera nécessaire, comme par exemple en utilisant des B-splines ou des réseaux de neurones.

Dans notre cas, il s'agit bien sûr d'approximer les coefficients de forces et de moments aérodynamiques sous la forme de fonctions analytiques différentiables. Nous avons obtenu ces tables de coefficients à partir de la base aérodynamique de l'orbiteur STS-1, contenue dans un manuel de plus de 1600 pages [166]. Nous avons alors envisagé trois techniques permettant d'approximer ces coefficients :

- La première solution consiste à effectuer une *analyse en composantes principales* en dimension  $n$  grâce à un ensemble de décompositions en valeurs singulières, puis à approximer les vecteurs propres à gauche et à droite par des polynômes de degrés réduits,
- La seconde solution utilise la propriété de parcimonie des *réseaux de neurones statistiques* afin d'approximer les coefficients sous la forme de fonctions non linéaires multivariées suffisamment lisses,
- La troisième solution consiste en une *hybridation* des deux techniques précédentes.

Cette dernière solution est décrite en détails dans la suite.

Le principe de la méthode d'approximation basée sur l'analyse en composantes principales est relativement simple, et à été introduit dans [174] afin de modéliser des tables de données bi-dimensionnelles. L'avantage de cette méthode est qu'elle permet de réaliser une *séparation des variables*, qui présentera un intérêt majeur lors de l'écriture de l'état et des commandes du modèle du véhicule en fonction des sorties plates et de leurs dérivées (voir section 3.3.2).

Plus précisément, toute matrice de taille  $m \times n$  à valeurs réelles peut se réécrire sous la forme d'une combinaison linéaire

$$A = \sum_{k=1}^r \sigma_k u_k v_k^T, \quad (3.21)$$

où les  $\sigma_k$  sont les valeurs singulières de  $A$  et  $r$  désigne le rang de la matrice. De plus,  $u_k$  et  $v_k$  représentent respectivement les vecteur singuliers à gauche et à droite.

**Remarque 3.10.** En pratique, on choisit lorsque cela est possible un ordre de troncature approprié  $p \leq r$  de la décomposition en valeurs singulières (3.21) en analysant le niveau d'énergie de chaque composante, de manière à simplifier la forme analytique finale tout en perdant le moins d'informations possible.

Alors, si on considère que l'on possède au départ un ensemble de points discrets  $(x_i, y_j, A_{ij})$ ,  $i = 1, \dots, m$ ,  $j = 1, \dots, n$ , le processus suivant est utilisé :

1. Calculer la décomposition en valeurs singulières de la matrice  $A$ ,
2. Approximer les  $r$  ensembles de points  $(x_i, (u_k)_i)$ ,  $1 \leq i \leq m$  (associés aux vecteurs singuliers à gauche  $u_k$ ) par des fonctions différentiables  $u_k(x)$ ,

3. Approximer les  $r$  ensembles de points  $(y_j, (v_k)_j)$ ,  $1 \leq j \leq n$  (associés aux vecteurs singuliers à droite  $v_k$ ) par des fonctions différentiables  $v_k(y)$ ,

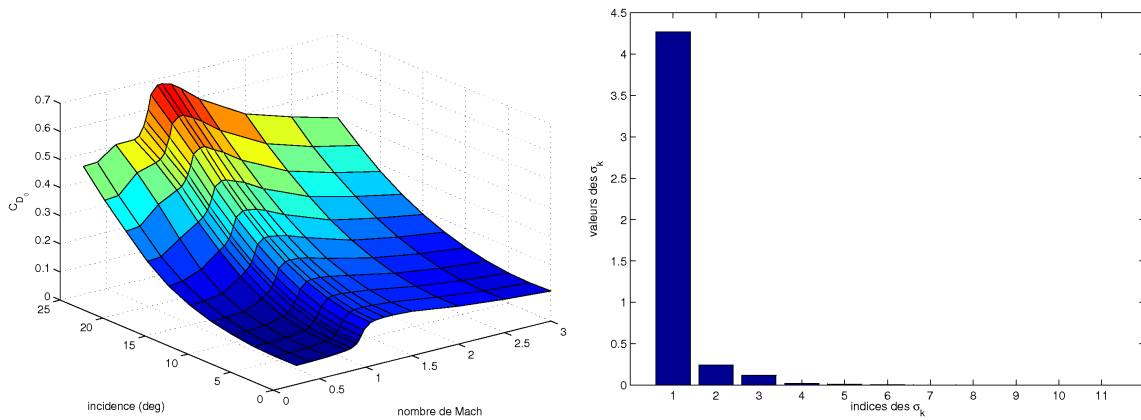
La fonction analytique  $\zeta$  résultant de ce processus constitue une approximation différentiable parcimonieuse de l'ensemble de points initial  $(x_i, y_j, A_{ij})$ . Cette dernière s'écrit alors

$$\zeta(x, y) = \sum_{k=1}^r \sigma_k u_k(x) v_k(y). \quad (3.22)$$

En d'autres termes,  $\zeta(x_i, y_j) = \sum_{k=1}^r \sigma_k u_k(x_i) v_k(y_j)$  approxime la valeur désirée

$$A_{ij} = \sum_{k=1}^r \sigma_k u_{ik} v_{jk}.$$

Illustrons la méthodologie précédente avec le coefficient de traînée de l'orbiteur STS-1 en configuration lisse, décrit dans la base aérodynamique [166] par une matrice  $A_{CD}$  de dimension  $11 \times 17$ , paramétrée en fonction de l'angle d'incidence  $\alpha$ , variant dans l'intervalle  $[0^\circ, 25^\circ]$ , et du nombre de Mach variant entre 0 et 3 (dans le cas des phases TAEM et A&L). Ce coefficient est représenté sur la figure 3.6(a). En effectuant une décomposition en valeurs singulières de la matrice  $A_{CD}$ , on obtient les valeurs singulières représentées sous forme décroissante sur l'histogramme 3.6(b). Clairement, la première valeur singu-



(a) Coefficient de traînée tabulé en configuration lisse. (b) Valeurs singulières associées au coefficient de traînée en configuration lisse.

lière d'amplitude 4.3 sur la figure 3.6(b) est au moins 13 fois plus grande que les valeurs singulières d'ordres plus élevés. Par conséquent, il est possible ici de tronquer la décomposition SVD précédente à l'ordre 1, c'est à dire en que l'on pose  $r = 1$  dans l'expression (3.22). Dans ce cas, les vecteurs singuliers à gauche et à droite associés à la plus grande valeur singulière sont représentés sur la figure 3.6 en fonction des variables indépendantes (angle d'incidence et nombre de Mach). Lorsque les vecteurs singuliers à gauche et à droite

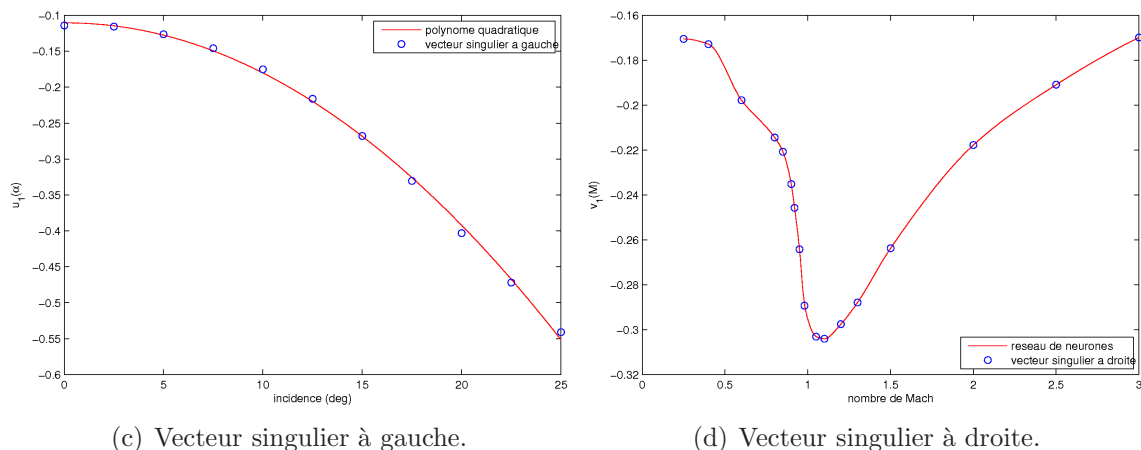


FIG. 3.6 – vecteurs singuliers à gauche et à droite associés à la plus grande valeur singulière de  $A_{CD}$ .

sont suffisamment lisses, il est possible d'utiliser des techniques d'interpolation élémentaires. Par exemple, le vecteur singulier à gauche peut être facilement approximé par un polynôme d'ordre 2 (voir figure 3.6(c)) en fonction de l'angle d'incidence, tel que

$$u_1(\alpha) = -0.7 \cdot 10^{-3} \alpha^2 + 0.15 \cdot 10^{-3} \alpha - 0.11. \quad (3.23)$$

En revanche, l'utilisation de méthodes plus complexes peut devenir nécessaire afin d'éviter toute perte d'information. C'est notamment le cas du vecteur singulier à droite  $v_1$  représenté sur la figure 3.6(d).

Il est alors possible d'utiliser des réseaux de neurones artificiels (ANN) statiques, tels que les perceptrons multi-couches (MLP). Ces réseaux sont constitués d'un ensemble de neurones distribués dans un certain nombre de couches, communément une couche d'entrée, une série de couches cachées, et une couche de sortie. De manière générale, les réseaux de neurones sont capables d'approximer des relations non linéaires très complexes entre des variables d'entrée et des variables de sortie. En outre, les réseaux de neurones non bouclés à apprentissage supervisé sont bien connus pour être des approximateurs universels parcimonieux [65, 66].

Dans le cadre de cette étude, nous considérons que les fonctions d'activation des neurones du réseau MLP sont définies par une fonction de transfert de type log-sigmoïde<sup>56</sup>, et que les neurones de sortie sont activés par une fonction de transfert linéaire. le nombre de neurones cachés est choisi de manière heuristique afin de réaliser un compromis entre la qualité de l'apprentissage réalisé et la capacité de généralisation du réseau, tout en évitant d'obtenir des modèles sous-ajustés ou sur-ajustés.

<sup>56</sup>une fonction de transfert de type log-sigmoïde est de la forme  $f(x) = \frac{1}{1 + e^{-x}}$ .

Afin de construire la relation non linéaire entrées-sorties, le processus d'obtention du réseau commence par une *phase d'apprentissage*, dont l'objectif est de déterminer l'ensemble des paramètres optimaux du réseau  $\hat{\theta}$  (poids et biais) qui minimisent une fonction de coût prédéfinie  $J_{N_a}(\theta, Z^{N_a})$ , représentative des erreurs entre les sorties initiales mesurées  $y(k)$  et les sorties  $\hat{y}$ , prédites par le réseau. Par exemple, un critère classique aux moindres carrés, défini par

$$J_{N_a}(\theta, Z^{N_a}) = \frac{1}{2N_a} \cdot \sum_{k=1}^{N_a} [y(k) - \hat{y}(k/\theta)]^T [y(k) - \hat{y}(k/\theta)] ,$$

peut être utilisé, où  $Z^{N_a} = \{[u(k), y(k)]_{|k=1, \dots, N_a}\}$  représente un ensemble de *données d'apprentissage*, constitué de  $N_a$  entrées  $u(k)$  et des sorties correspondantes  $y(k)$ . Finalement, en utilisant un algorithme itératif de descente de gradient (Levenberg-Marquardt [175, 176], rétro-propagation, moindres carrés généralisés itératifs, etc.), les paramètres optimaux du réseau sont calculés sous la forme

$$\hat{\theta} = \operatorname{argmin} J_{N_a}(\theta, Z^{N_a})$$

Une fois la phase d'apprentissage du réseau terminée, une *phase de validation* (également appelée *phase de généralisation*) est nécessaire afin de vérifier si, en fonction de la qualité de l'apprentissage réalisé et de sa configuration actuelle (nombre de neurones, durée de l'apprentissage), le réseau est capable de traiter des données inconnues. Ainsi, l'objectif principal est de trouver le modèle le plus parcimonieux [66] fournissant un compromis entre les deux *indicateurs de performance* suivants :

- (a) l'erreur quadratique moyenne d'apprentissage (LMSE), définie par

$$LMSE = \sqrt{\frac{1}{N_a} \times \sum_{k=1}^{N_a} [y(k) - \hat{y}(k/\theta)]^2}.$$

- (b) l'erreur quadratique moyenne sur l'ensemble de test (TMSE), telle que

$$TMSE = \sqrt{\frac{1}{N_t} \times \sum_{k=1}^{N_t} [y(k) - \hat{y}(k/\theta)]^2}.$$

où  $N_a$  et  $N_t$  représentent respectivement le nombre de données d'apprentissage du réseau et le nombre d'échantillons dédiés à l'étape de validation.

Reprenons maintenant l'exemple associé à l'approximation du coefficient de traînée  $C_{D_0}$ . Nous avons choisi d'utiliser un réseau de neurones MLP comprenant un neurone d'entrée, une couche cachée comportant cinq neurones, et un neurone de sortie. L'optimisation du réseau se termine après 165 époques, et les indicateurs LMSE et TMSE sont tous les deux égaux à  $6.8^{-4}$ , ce qui indique que le réseau possède une très bonne capacité de généralisation. L'expression de l'approximation du vecteur propre à droite  $v_1$  en fonction du nombre

de Mach est alors donnée par

$$v_1(M) = \frac{0.08}{1 + \exp^{-2.2M+5.6}} - \frac{0.04}{1 + \exp^{5.6M-9.9}} + \frac{0.05}{1 + \exp^{-7.7M+10.3}} \quad (3.24)$$

$$+ \frac{0.097}{1 + \exp^{29.6M-27.9}} - \frac{0.05}{1 + \exp^{-11.4M+6.5}} - 0.22.$$

La courbe correspondante est représentée sur la figure 3.6(d). Finalement, en notant que la première valeur singulière vaut  $\sigma_1 = 4.26$ , l'expression analytique de  $C_{D_0}$  est donnée par

$$C_{D_0}(\alpha, M) = \sigma_1 u_1(\alpha) v_1(M), \quad (3.25)$$

où  $u_1(\alpha)$  et  $v_1(M)$  sont définis par (3.23) et (3.24) respectivement.

Finalement, la figure 3.7 montre l'erreur d'approximation entre les données originales et l'approximation du coefficient de traînée donnée par l'expression (3.25). La méthode pré-

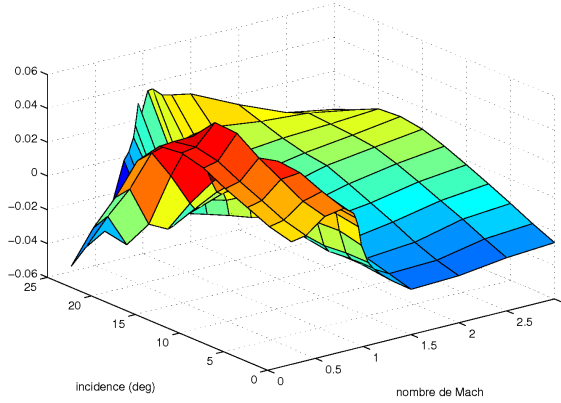


FIG. 3.7 – erreur d'approximation du coefficient de traînée en configuration lisse.

cédente peut également être utilisée pour approximer des données discrètes de dimension supérieure à 2, tout en préservant la propriété de découplage des variables indépendantes. Dans, ce cas, il est nécessaire d'effectuer des décompositions en valeurs singulières de manière itérative. Par exemple, supposons que nous cherchons à modéliser un ensemble de valeurs discrètes définies par  $(x_i, y_j, z_k, A_{ijk}), i = 1, \dots, m, j = 1, \dots, n, k = 1, \dots, o$ . Pour chaque valeur de  $k$  fixée, nous pouvons alors décomposer la matrice  $A_{ijk}$  de dimension  $m \times n$  en valeurs singulières sous la forme

$$\xi(x_i, y_j, z_k) = \sum_{p=1}^{\alpha} \sigma_p u_p(x_i, z_k) v_p(y_j, z_k). \quad (3.26)$$

Ensuite, en renouvelant la même opération pour les vecteurs propres à gauche et à droite, on obtient

$$u_p(x_i, z_k) = \sum_{q=1}^{\beta} \sigma'_{pq} u'_{pq}(x_i) v'_{pq}(z_k) \quad (3.27)$$

et

$$v_p(y_j, z_k) = \sum_{r=1}^{\gamma} \sigma_{pr}'' u_{pr}''(y_j) v_{pr}''(z_k). \quad (3.28)$$

En réinjectant les expressions (3.27) et (3.28) dans (3.26), il vient

$$\xi(x_i, y_j, z_k) = \sum_{p=1}^{\alpha} \sum_{q=1}^{\beta} \sum_{r=1}^{\gamma} \sigma_p \sigma'_{pq} \sigma''_{pr} u'_{pq}(x_i) u''_{pr}(y_j) v''_{pr}(z_k) v'_{pq}(z_k).$$

Finalement, en approximant les données discrètes  $u'_{pq}(x_i)$ ,  $u''_{pr}(y_j)$ ,  $v''_{pr}(z_k)$  et  $v'_{pq}(z_k)$  par des fonctions continues, on obtient une expression analytique différentiable, telle que

$$\xi(x, y, z) = \sum_{p=1}^{\alpha} \sum_{q=1}^{\beta} \sum_{r=1}^{\gamma} \sigma_p \sigma'_{pq} \sigma''_{pr} u'_{pq}(x) u''_{pr}(y) v''_{pr}(z) v'_{pq}(z).$$

La démarche précédente se généralise facilement en dimension  $n$ .

Lorsque la propriété de séparation des variables n'est pas recherchée, il peut être intéressant d'approximer certains coefficients aérodynamiques sous forme analytique différentiable en utilisant uniquement des réseaux de neurones statiques. En effet, de part leur propriété de parcimonie, les relations non linéaires obtenues comporteront alors un faible nombre de termes. Par exemple, la figure 3.8 illustre l'approximation du coefficient de roulis de l'orbiteur STS-1 dû à la gouverne de direction, réalisée en utilisant un perceptron multi-couches uniquement.

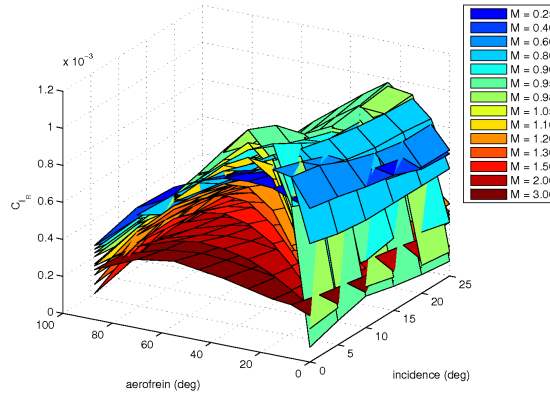


FIG. 3.8 – contribution de la gouverne de direction sur le coefficient de moment de lacet.

### 3.3.2 Planification de trajectoires TAEM par platitude

Dans cette section, nous allons utiliser la propriété de platitude du modèle à trois degrés de liberté du véhicule (3.3)-(3.8) afin de transcrire le problème de commande optimale (3.15) associé à la phase TAEM dans l'espace des sorties plates. En ce qui concerne

la planification de trajectoires TAEM, nous proposons deux formulations du problème, basées sur un changement de variable indépendante dans les équations (3.3)-(3.8). En effet, le temps ne représente pas une variable pertinente pour le problème de planification de trajectoires de rentrée atmosphérique, car nous ne cherchons pas à atteindre le point ALI en un temps prédéfini. La première formulation proposée utilise un facteur d'échelle  $\lambda$  qui va venir moduler la durée totale de la trajectoire TAEM. La seconde formulation, plus classique dans le cadre des problèmes de planification de trajectoires de rentrée atmosphérique, utilise l'énergie totale du véhicule comme variable indépendante du problème. En effet, ce choix semble être le plus naturel car le but de la phase TAEM consiste à dissiper un niveau d'énergie donné entre le point TEP et l'interface avec la phase A&L.

### Première formulation : trajectoire TAEM de durée libre

Nous allons à présent utiliser la sortie plate, déterminée par calcul formel dans la proposition 3.1, afin d'inverser le modèle 3 ddl du véhicule de manière analytique, et de réécrire l'état et les commandes du système en fonction de cette sortie plate et de ses dérivées. Dans le cadre du guidage pour les véhicules de rentrée atmosphérique, l'inversion par platitude du modèle couplé ne nécessite pas de poser l'hypothèse classique de découplage du mode longitudinal et du mode latéral, comme cela est souvent fait en pratique dans les stratégies de guidage linéaires. Cette propriété constitue ainsi un résultat important, permettant de réduire un certain degré de conservatisme inhérent aux méthodes classiques.

En choisissant  $(x, y, h)$  comme sortie plate candidate (voir la proposition 3.1), l'état  $(x, y, h, V, \gamma, \chi)$  et les commandes  $(\alpha, \beta, \mu)$  du modèle (3.3)-(3.8) peuvent être réécrits exclusivement en fonction de cette sortie plate et d'un nombre fini de ses dérivées. Cependant, il est important de pouvoir générer des trajectoires équilibrées afin que le degré de liberté associé à l'angle de dérapage puisse être utilisé par le régulateur de poursuite de trajectoire, ainsi que par la loi de pilotage, afin de contrer certaines perturbations aérodynamiques et certains défauts d'équilibrage du véhicule. Ainsi, nous supposons que l'angle de dérapage  $\beta$  est nul, et que des conditions de vol symétrique existent. Par conséquent, la force latérale  $Y$  est nulle. Finalement, nous remplaçons la variable temporelle du modèle (3.3)-(3.8) par une durée normalisée, ajustée par un scalaire  $\lambda$ .

En introduisant la contrainte  $\beta = 0$  et la durée normalisée  $\tau = \frac{t}{\lambda}$ , avec  $0 \leq \tau \leq 1$ , alors on peut écrire

$$(\cdot)' = \frac{d(\cdot)}{d\tau} = \lambda \frac{d(\cdot)}{dt},$$



et les équations du modèle 3 ddl (3.3)-(3.8) deviennent

$$x' = \lambda V \cos \chi \cos \gamma \quad (3.29)$$

$$y' = \lambda V \sin \chi \cos \gamma \quad (3.30)$$

$$h' = \lambda V \sin \gamma \quad (3.31)$$

$$V' = \lambda \left( -\frac{D}{m} - g \sin \gamma \right) \quad (3.32)$$

$$\gamma' = \lambda \left( \frac{L \cos \mu}{mV} - \frac{g}{V} \cos \gamma \right) \quad (3.33)$$

$$\chi' = \lambda \frac{L \sin \mu}{mV \cos \gamma} \quad (3.34)$$

Le nouveau modèle (3.29)-(3.34) peut toujours s'écrire en fonction de la sortie plate  $z$ , constituée des variables de position du centre de gravité du véhicule dans le repère piste  $(X_{rwy}, Y_{rwy}, Z_{rwy})$ , i.e., la portée longitudinale  $x = z_1$ , la portée latérale  $y = z_2$  et l'altitude  $h = z_3$ . Les nouvelles consignes de guidage sont alors l'angle de gîte aérodynamique  $\mu$  et l'angle d'incidence  $\alpha$ . En fait, l'ensemble de l'état et des commandes du modèle non linéaire (3.29)-(3.34) peuvent se réécrire en fonction de  $z$ , de ses dérivées, ainsi que de la durée totale de la trajectoire TAEM représentée par le scalaire  $\lambda$ . Les expressions des états sont alors données par

$$V = \frac{\sqrt{z_1'^2 + z_2'^2 + z_3'^2}}{\lambda}, \quad (3.35)$$

$$\gamma = \arctan \left( \frac{z_3'}{\sqrt{z_1'^2 + z_2'^2}} \right), \quad (3.36)$$

$$\chi = \arctan \left( \frac{z_2'}{z_1'} \right), \quad (3.37)$$

La consigne de guidage  $\mu$  est ensuite calculée à partir des équations (3.33) et (3.34) telle que

$$\mu = \arctan \left( \frac{\chi' \cos \gamma}{\gamma' + \frac{g \cos \gamma}{V} \lambda} \right), \quad (3.38)$$

où

$$\chi' = \frac{z_2'' z_1' - z_2' z_1''}{z_1'^2 + z_2'^2} \quad (3.39)$$

et

$$\gamma' = \frac{z_3''(z_1'^2 + z_2'^2) - z_3'(z_1' z_1'' + z_2' z_2'')}{(z_1'^2 + z_2'^2 + z_3'^2) \sqrt{z_1'^2 + z_2'^2}}. \quad (3.40)$$

Afin d'inverser l'équation (3.34) pour obtenir l'expression analytique associée à l'angle d'incidence  $\alpha$ , nous commençons par modéliser le coefficient de portance en configuration lisse  $C_{L_0}(\alpha, M)$  en utilisant l'approche par séparation de variables présentée dans la section 3.3.1, sous la forme  $\sigma_1 u_1(\alpha) v_1(M)$ , telle que

$$C_{L_0}(\alpha, M) = (a_0 + a_1 \alpha) f_{C_{L_0}}(M), \quad (3.41)$$

où  $f_{C_{L_0}}(M)$  est un réseau de neurones statique semblable à (3.24).

Finalement, en réinjectant l'expression (3.41) dans l'équation (3.33), et en notant que  $L = \bar{q}SC_{L_0}(\alpha, M)$ , on obtient

$$\alpha = \frac{2m}{a_1 f_{C_{L_0}}(M) \rho S V \cos \mu} \left( \frac{\gamma'}{\lambda} + \frac{g \cos \gamma}{V} \right) - \frac{a_0}{a_1}, \quad (3.42)$$

qui ne présente pas de singularité car  $-80^\circ \leq \mu \leq 80^\circ$  le long du segment de vol TAEM.

**Remarque 3.11.** L'expression (3.41) fait clairement apparaître l'expression linéaire  $a_0 + a_1 \alpha$  typiquement utilisée pour définir le coefficient de portance en configuration lisse, ainsi qu'un terme dépendant de  $M$ , obtenu grâce à un réseau de neurones, qui va influencer le profil du coefficient de portance principalement pour les nombre de Mach proches du transsonique (voir figure 3.9).

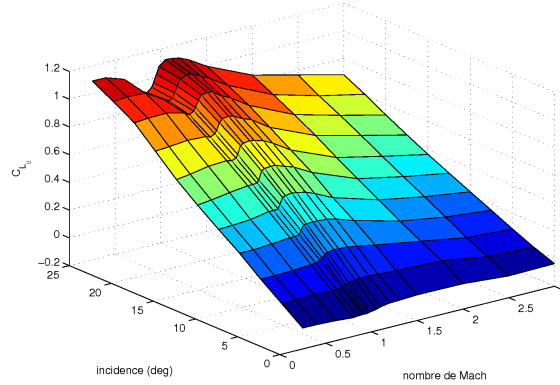


FIG. 3.9 – coefficient de portance tabulé en configuration lisse.

Cependant, le modèle (3.29)-(3.34) n'est plus plat pour  $\beta = 0$ , et présente un observable autonome. Ainsi, il est nécessaire que l'équation (3.32) soit strictement vérifiée, c'est à dire :

$$\Lambda_\tau(x, u) = \frac{V'}{\lambda} + g \sin \gamma + \frac{1}{2} \frac{\rho S V^2 C_{D_0}(\alpha, M)}{m} = 0, \quad (3.43)$$

où chacune des variables ( $V, V', \gamma, V, \dots$ ) intervenant dans la contrainte égalitaire (3.43) peuvent s'écrire exclusivement en fonction de la sortie plate, de ses dérivées, ainsi que de la durée totale de la trajectoire  $\lambda$ . En particulier,  $V'$  s'écrit dans l'espace des sorties plates et de leurs dérivées sous la forme

$$V' = \frac{z'_1 z''_1 + z'_2 z''_2 + z'_3 z''_3}{\lambda \sqrt{z_1^2 + z_2^2 + z_3^2}}.$$

**Remarque 3.12.** Le principe de séparation des variables n'étant pas nécessaire pour écrire l'équation (3.43), le coefficient de traînée en configuration lisse  $C_{D_0}$  est modélisé en utilisant uniquement un réseau de neurones statique.

**Remarque 3.13.** Dans les équations (3.42) et (3.43), la densité atmosphérique s'écrit simplement en fonction de  $z_3$  sous la forme

$$\rho = \rho_0 \exp(-z_3/H_0).$$

Finalement, les contraintes mécaniques de trajectoires, c'est-à-dire la pression dynamique  $\bar{q}$  définie par (3.12) et la contrainte de facteur de charge (3.14) peuvent également s'écrire en fonction des sorties plates, de leurs dérivées, ainsi que de la durée totale de la trajectoire définie par le scalaire  $\lambda$ .

En utilisant les relations d'équivalence (2.5), c'est-à-dire les expressions des états et des commandes définies respectivement par (3.35)-(3.37) et (3.38)-(3.42), nous sommes maintenant en mesure de réécrire le problème de commande optimale (3.15), associé à la planification de trajectoires TAEM, dans l'espace des sorties plates. On obtient alors le problème de faisabilité

$$\text{trouver } (\bar{z}(t), \lambda) \tag{3.44}$$

t.q.

$$\begin{aligned} & \psi_x(\bar{z}(\tau_0), \lambda) = x_0, \\ & \psi_u(\bar{z}(\tau_0), \lambda) = u_0, \\ & \Lambda_\tau(\psi_x(\bar{z}(\tau), \lambda), \psi_u(\bar{z}(\tau), \lambda)) = 0, \quad \tau \in [\tau_0, \tau_f], \\ 0 & \leq \Gamma(\psi_x(\bar{z}(\tau), \lambda), \psi_u(\bar{z}(\tau), \lambda)) \leq \Gamma_{\max}, \quad \tau \in [\tau_0, \tau_f], \\ 0 & \leq \bar{q}(\psi_x(\bar{z}(\tau), \lambda)) \leq \bar{q}_{\max}, \quad \tau \in [\tau_0, \tau_f], \\ u_{\min} & \leq \psi_u(\bar{z}(\tau), \lambda) \leq u_{\max}, \quad \tau \in [\tau_0, \tau_f], \\ & \psi_x(\bar{z}(\tau_f), \lambda) = x_f, \\ & \psi_u(\bar{z}(\tau_f), \lambda) = u_f, \end{aligned}$$

où le vecteur  $\bar{z}$  est défini par  $\bar{z} = (z_1, z_2, z_3, \dot{z}_1, \dot{z}_2, \dot{z}_3, \ddot{z}_1, \ddot{z}_2, \ddot{z}_3)$  dans l'espace des jets d'ordre infini, et où le temps normalisé  $\tau$  vérifie  $\tau_0 = 0$  et  $\tau_f = 1$ . De plus, la contrainte égalitaire de trajectoire  $\Lambda_\tau$  est définie par l'équation autonome (3.43).

Afin de résoudre le problème de commande optimale (3.44), il est nécessaire de le transformer en un problème de programmation non linéaire. Comme indiqué dans la section 2.6, nous paramétrons alors les sorties plates  $z_1(t)$ ,  $z_2(t)$  et  $z_3(t)$  par des courbes B-splines, puis nous discrétisons le problème afin de faire apparaître les paramètres de contrôle des B-splines qui deviennent alors les variables de décision du problème NLP.

Nous avons paramétré les sorties plates  $z_1(t)$ ,  $z_2(t)$  et  $z_3(t)$  par des B-splines d'ordre 6, de multiplicité 4 et comportant 6 intervalles, afin d'obtenir des dérivées temporelles d'ordre 1 et d'ordre 2 continues. Le scalaire  $\lambda$  associé à la durée totale de la trajectoire est paramétré par une B-spline d'ordre 1, de multiplicité 0 et comportant 1 intervalle. De plus,  $\lambda$  est autorisé à varier entre 50 et 500 s, ce qui est largement suffisant pour pouvoir décrire l'ensemble des trajectoires TAEM potentielles solutions du problème (3.44). Les contraintes de trajectoire sont vérifiées en 24 points de collocation, uniformément répartis entre  $\tau_0$  et  $\tau_f$ .

**Remarque 3.14.** Comme nous nous sommes focalisés ici sur la détermination de trajectoires TAEM admissibles en fonction d’un certain nombre d’incertitudes cinématiques initiales, nous n’avons pas considéré de fonctionnelle de coût particulière dans le problème de commande optimale (3.44). Bien sûr, nous aurions pu chercher à planifier des trajectoires de rentrée à consommation de gîte minimale, ou encore réduisant au maximum le déport latéral du véhicule le long de la trajectoire.

Nous avons effectué une campagne de simulations Monte Carlo en considérant un ensemble de dispersions initiales au point d’entrée TEP (portée longitudinale, portée latérale, altitude, pente et cap) dues aux incertitudes et imperfections associées à la loi de guidage hypersonique. Seule la vitesse relative du véhicule n’est pas dispersée, la transition entre la phase hypersonique et la phase TAEM, et donc le “basculement” de la loi de guidage dans le calculateur de vol, étant typiquement effectuée sur un critère de nombre de Mach. Les conditions cinématiques nominales, ainsi que les dispersions considérées, sont reportées dans le tableau 3.6. En outre, les conditions finales considérées au point ALI sont indiquées dans le tableau 3.3. De plus, la figure 3.10 illustre le plan d’expérience

paramètre	symbole	valeur nominale	$\sigma$
<i>Position</i>			
portée longitudinale initiale [km]	$x_0$	-20	$\pm 7$
portée latérale initiale [km]	$y_0$	-30	$\pm 7$
altitude initiale [km]	$h_0$	25	$\pm 3$
<i>Vitesse</i>			
nombre de Mach initial <sup>57</sup>	$M_0$	2	N.A.
pente initiale [deg]	$\gamma_0$	-5	$\pm 2$
cap initial [deg]	$\chi_0$	-30	$\pm 10$

TAB. 3.6 – conditions initiales considérées pour l’optimisation de trajectoires TAEM.

utilisé pour partitionner le domaine de positions initiales du centre de gravité du véhicule au point TEP. Une démarche similaire est utilisée pour quadriller tout l’espace de dimension 5 défini par les paramètres  $(x_0, y_0, h_0, \gamma_0, \chi_0)$ . En choisissant huit valeurs discrètes de positions initiales du centre de masse du véhicule sur chaque axe, ainsi que trois valeurs discrètes de pentes et de caps initiaux, on obtient alors un total de 523 cas de simulations Monte Carlo. Finalement, les paramètres du modèle exponentiel de densité atmosphérique  $\rho = \rho_0 \exp(-h/H_0)$  sont choisis tels que  $\rho_0 = 1.293 \text{ kg/m}^3$  et  $H_0 = 7100 \text{ m}$ . L’ensemble des optimisations de trajectoires a été réalisé sous l’environnement Matlab à l’aide d’un ordinateur personnel équipé d’un processeur Intel Core 2 Duo cadencé à 2 GHz. Le solveur de problèmes de programmation non linéaire qui a été utilisé pour cette étude est NPSOL [44].

La figure 3.11 illustre les trajectoires TAEM obtenues dans un repère cartésien local

<sup>57</sup>le nombre de Mach étant simplement lié à la vitesse relative du véhicule par la relation  $M = V/V_s$ , où la vitesse moyenne du son  $V_s$  aux altitudes considérées est choisie telle que  $V_s = 305.8 \text{ m/s}$ , la condition initiale  $M_0$  est donc équivalente la définition d’une vitesse relative initiale  $V_0$ .

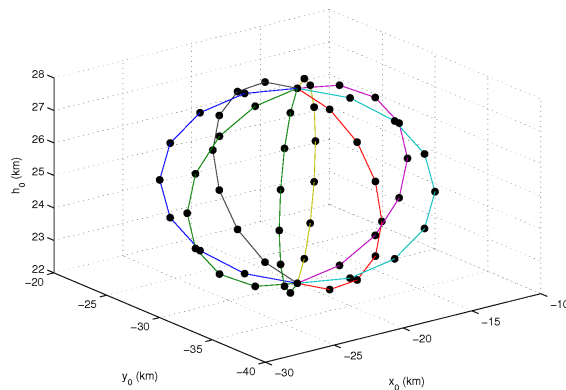


FIG. 3.10 – discrétisation des conditions cinématiques initiales de position.

attaché au centre de la piste. Toutes les optimisations effectuées lors de cette simulation Monte Carlo ont convergé vers une solution optimale. Un estimé initial de la trajectoire TAEM, correspondant au cas nominal, a été utilisée afin d'initialiser le solveur NPSOL. Bien sûr, les solveurs de type NLP étant relativement sensibles au choix de l'estimé initial, une augmentation trop importante du domaine de dispersions cinématiques initiales résulterait en une diminution de la robustesse du solveur, c'est à dire en une perte potentielle de convergence du problème d'optimisation pour des conditions cinématiques initiales trop éloignées de l'estimation initiale.

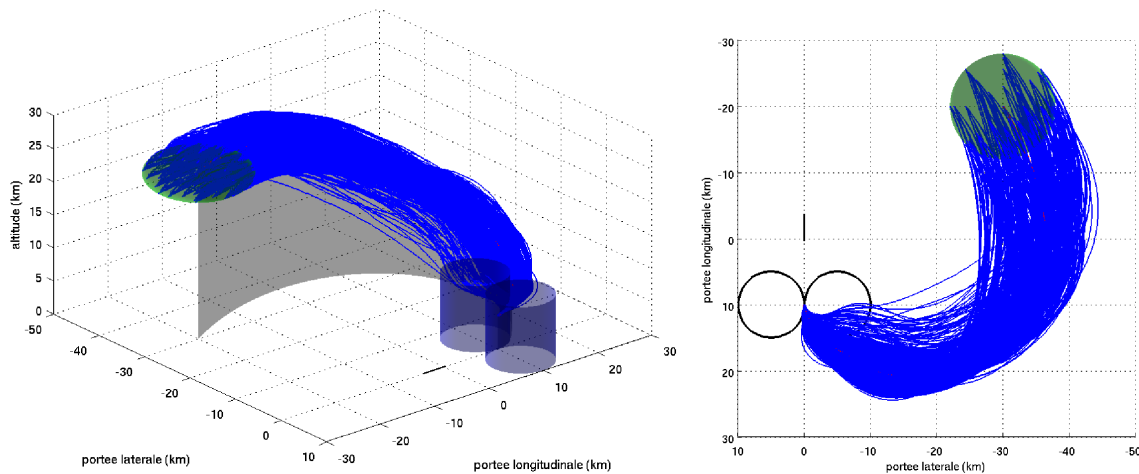


FIG. 3.11 – trajectoires TAEM en présence de dispersions cinématiques initiales.

**Remarque 3.15.** Les cylindres d'alignement HAC sont représentés uniquement à titre indicatif sur la figure 3.11. Ils n'interviennent absolument pas dans le problème de commande optimale (3.44). Contrairement aux techniques d'optimisation basées sur un partitionnement de la trajectoire TAEM en un petit nombre de segments géométriques, la

méthode proposée à l'avantage de ne pas nécessiter l'utilisation de paramètres de réglage exogènes. Par exemple, pour un segment de trajectoire circulaire traduisant l'enroulement du véhicule de rentrée autour du cylindre d'alignement de cap, il serait alors également nécessaire d'optimiser les paramètres de dimensionnement de ce cylindre, c'est à dire sa position et son rayon.

De plus la fonction de densité de probabilité associée à la durée totale de trajectoire (paramètre  $\lambda$ ) est représentée sur la figure 3.12. Clairement, les bornes  $50 \text{ s} \leq \lambda \leq 500 \text{ s}$  introduites dans le problème de commande optimale ne sont jamais dépassées.

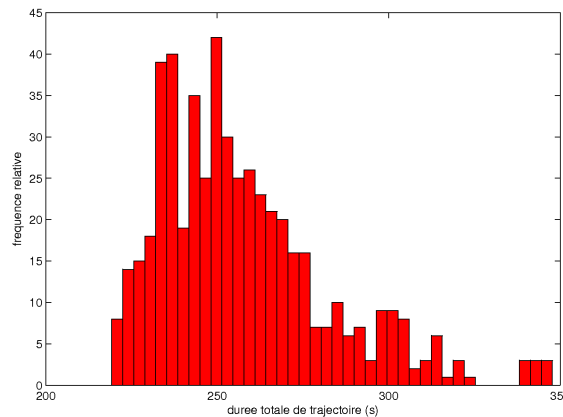


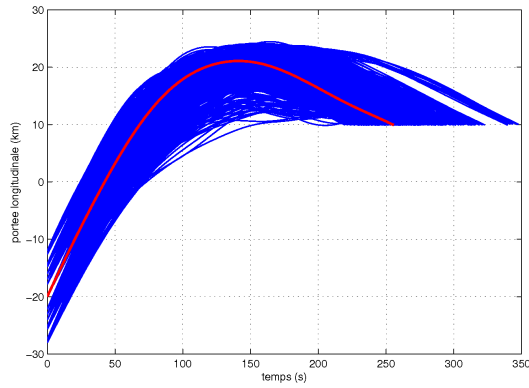
FIG. 3.12 – densité de probabilité associée à la durée des trajectoires TAEM obtenues.

En utilisant la propriété de platitude du modèle, on obtient ainsi des trajectoires TAEM sur lesquelles la dynamique du modèle 3 dll du véhicule est vérifiée de manière *exacte*. Une fois que les trajectoires optimales des sorties plates sont obtenues, les courbes B-splines sont dérivées de manière analytique, puis les relations d'équivalence (2.5) sont alors utilisées pour déterminer formellement les profils des consignes de guidage. Dans notre cas, les trajectoires de l'état du système peuvent être calculées grâce aux relations (3.35), (3.36) et (3.37), les profils de portée longitudinale, de portée latérale et d'altitude étant trivialement obtenus via les sorties plates. Ces derniers sont représentés sur la figure 3.13. Dans chaque cas, les profils nominaux associés à la trajectoire non dispersée<sup>58</sup>, sont représentés en rouge sur les figures suivantes.

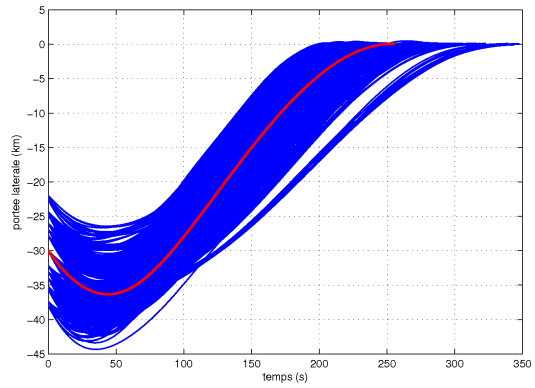
**Remarque 3.16.** Nous avons ajouté une contrainte supplémentaire de pente négative dans le problème de commande optimale. Cette contrainte est assez bien vérifiée (voir figure 3.13(e)), excepté quelques dépassements mineurs entre certains points de collocation liés à la méthode de transcription directe de l'OCP.

Ensuite, les consignes de guidage  $\alpha$  et  $\mu$  associées à ces différentes trajectoires TAEM peuvent être calculées grâce aux relations (3.38) et (3.38). nous obtenons alors les profils d'incidence et de gîte représentés sur les figures (3.14(a)) et (3.14(b)). En dehors de

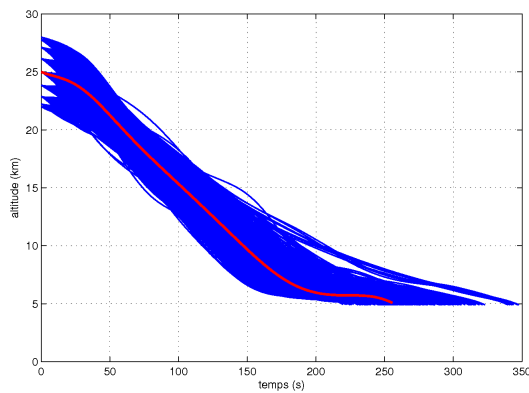
<sup>58</sup>qui joue également le rôle d'estimé initial pour le problème NLP



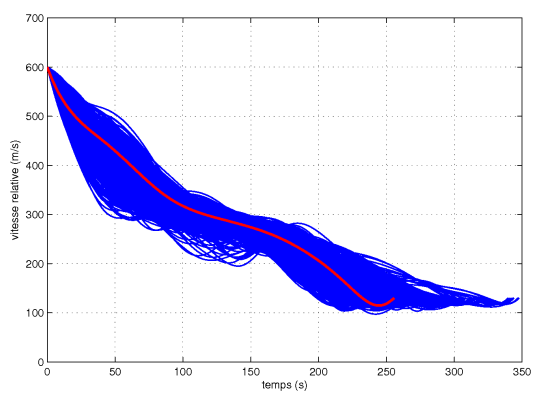
(a) Profils de portées longitudinales.



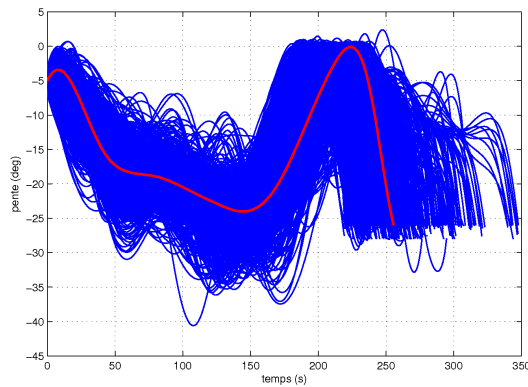
(b) Profils de portées latérales.



(c) Profils d'altitudes.



(d) Profils de vitesses relatives.



(e) Profils de pentes.

FIG. 3.13 – profils des états de référence.

quelques dépassements de faible amplitude en dehors des points de collocation (où les contraintes sont vérifiées), les bornes sur les ordres de guidage définies dans le tableau 3.4 sont globalement bien respectées.

En outre, les contraintes mécaniques de trajectoire (pression dynamique et facteur de

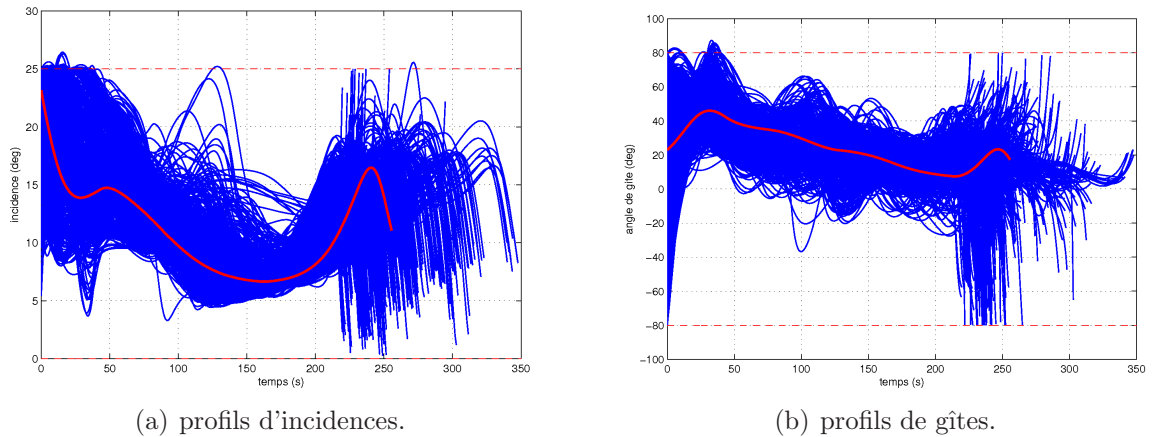


FIG. 3.14 – profils des consignes de guidage de référence.

charge), calculées en fonction des sorties plates grâce aux expressions (3.12) et (3.14), sont représentées sur les figures 3.15(a) et 3.15(b).

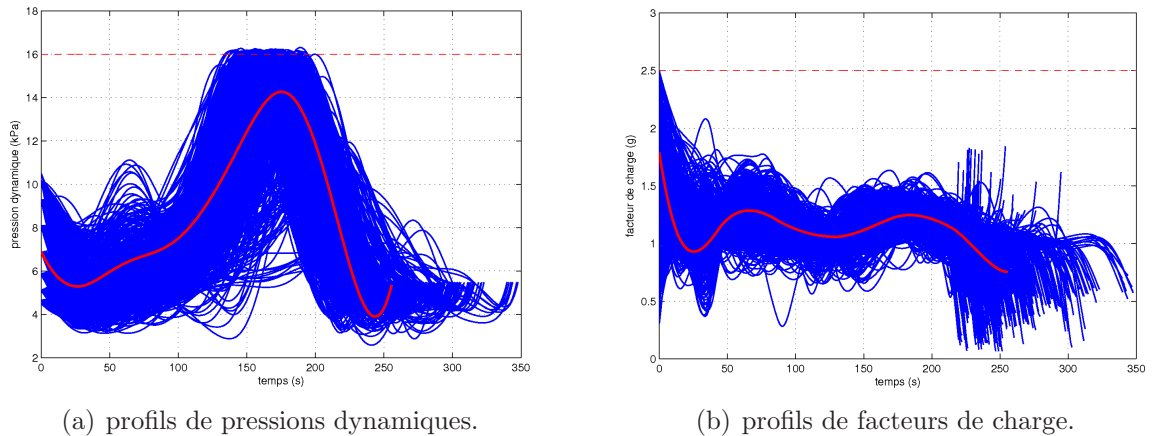


FIG. 3.15 – profils des contraintes mécaniques le long de la trajectoire de référence.

Le système (3.29)-(3.34) n'étant pas plat lorsque  $\beta = 0$ , l'équation égalitaire (3.43) doit être vérifiée le long de chaque trajectoire TAEM calculée. La figure 3.16(a) illustre la satisfaction de cette contrainte pour l'ensemble des trajectoires générées.

**Remarque 3.17.** L'espacement et le nombre de points de collocation doivent être choisis afin de minimiser les dépassements de la contrainte égalitaire (voir figure 3.16(a)). La satisfaction de la contrainte de dérapage nul peut ensuite être vérifiée *a posteriori* en réintégrant la dynamique non linéaire du véhicule à partir des profils optimaux des consignes de guidage.

Finalement, la figure 3.16(b) illustre le temps CPU nécessaire pour résoudre le problème de planification de trajectoires TAEM, qui est inférieur à la seconde dans tous les



cas, et inférieur ou égal à 50 ms pour 298 d'entre eux. Cependant, un certain nombre de

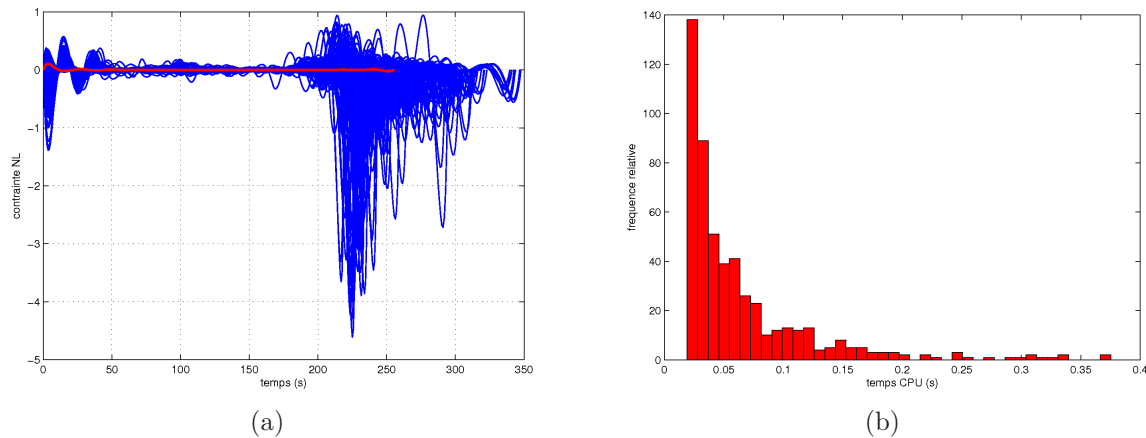


FIG. 3.16 – (a) contrainte liée à l'équation autonome du système ( $m/s^2$ ) (b) temps CPU nécessaire pour effectuer les différentes optimisations.

difficultés sont liées à la méthode de transcription directe utilisée pour résoudre le problème de commande optimale. par exemple, l'utilisation d'un ensemble de points discrets où sont vérifiées les contraintes de trajectoire du problème peut entraîner dans certains cas des violations de contraintes entre les points de collocation. C'est notamment le cas des petits dépassements observés sur plusieurs des figures précédentes (par exemple sur les figures 3.14(a), 3.14(b) ou 3.12). On pourrait alors quantifier le phénomène et décider d'une marge supplémentaire à appliquer sur les bornes des contraintes.

De plus, comme nous l'avons indiqué précédemment, les solveurs NLP sont relativement sensibles aux variations des conditions initiales. Aussi, de manière à agrandir le domaine de conditions cinématiques initiales couvert par l'algorithme de planification, une banque de trajectoires pourrait être générée afin de jouer le rôle d'estimé initial pour le solveur NLP. par exemple, un réseau de neurones statique pourrait apprendre les paramètres de contrôle des splines définissant cet ensemble discret de trajectoires. Le réseau serait ensuite interrogé à bord du véhicule et, en fonction des conditions initiales mesurées par le module de navigation, il pourrait alors fournir au solveur NLP la trajectoire connue la plus proche afin d'initialiser un processus d'optimisation local.

Pour conclure cette section, nous avons comparé les performances en temps de calcul CPU entre l'approche proposée et un certain nombre de méthodes de planification de trajectoires classiques implémentées dans l'outil logiciel ASTOS (AeroSpace Trajectory Optimization Software). Cet outil, utilisé à maintes reprises par l'Agence Spatiale Européenne (ESA/ESTEC) dans le cadre des programmes X-38 [177], Ariane [178], ATV [179] et, plus récemment, pour le lanceur Vega [180], est réputé pour la qualité des algorithmes d'optimisation implémentés. ASTOS utilise une méthode de collocation ou de tirs directs multiples pour résoudre les problèmes de planification de trajectoires. Nous avons alors cherché à résoudre le problème d'optimisation de trajectoires TAEM défini par (3.15),

pour des conditions initiales non dispersées, en utilisant une technique de collocation basée sur des polynômes de Hermite. Le problème est ensuite résolu par le solveur SNOPT [43]. Le modèle aérodynamique du véhicule (orbiteur STS-1), les modèles atmosphériques (densité atmosphérique de type exponentielle et champ de gravitation constant), le modèle terrestre (Terre sphérique non tournante), les conditions initiales et finales de trajectoire, ainsi que le nombre de contraintes de trajectoire utilisées sont rigoureusement les mêmes que précédemment. Comme précédemment, nous n'avons pas imposé d'indice de performance à minimiser le long de la trajectoire. La trajectoire TAEM optimale obtenue est représentée sur la figure 3.17, où le bandeau de couleur sur la partie droite représente le facteur de charge total le long de la trajectoire. Les profils des angles d'incidence et de

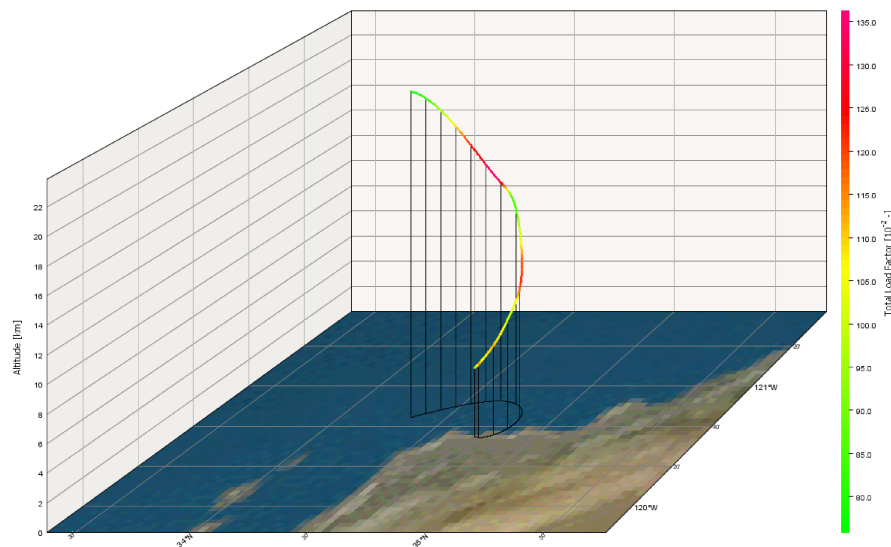
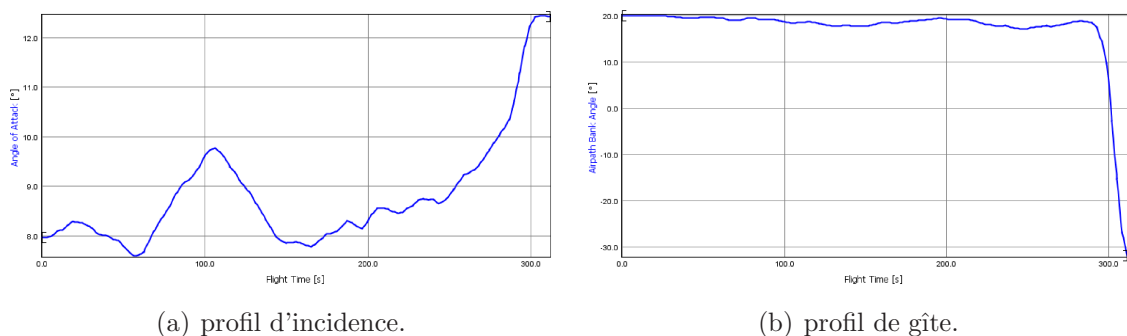


FIG. 3.17 – trajectoire TAEM obtenue sous l'environnement ASTOS.

gîte correspondant sont représentés sur les figures 3.18(a) et 3.18(b). De plus, les figures



(a) profil d'incidence.

(b) profil de gîte.

FIG. 3.18 – profils des commandes de référence obtenus sous l'environnement ASTOS.

3.19(a) et 3.19(b) illustrent les profils de pression dynamique et de facteur de charge obtenus.

nus le long de la trajectoire, associés aux contraintes mécaniques agissant sur le véhicule. L'optimisation de cette trajectoire TAEM en utilisant l'outil ASTOS a été effectuée en

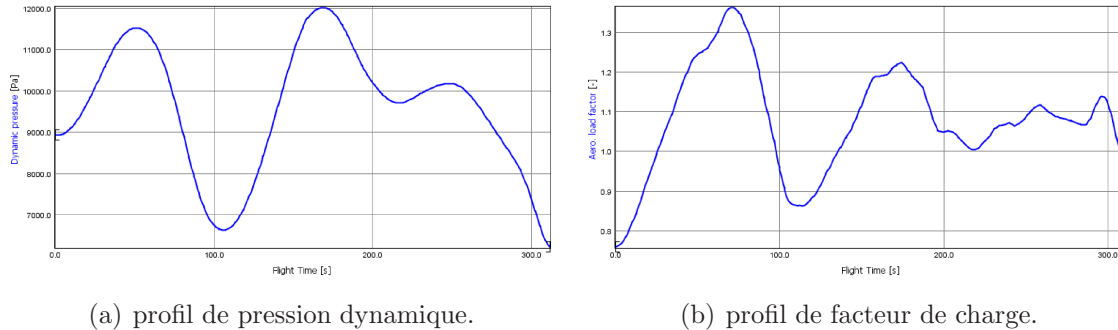


FIG. 3.19 – contraintes mécaniques obtenues sous l'environnement ASTOS.

36.3 s (temps CPU à comparer aux 0.37 s dans le pire cas pour la méthode proposée). La solution optimale est obtenue après 25 itérations du solveur SNOPT, en utilisant le même ordinateur personnel que précédemment. Comparé au temps CPU indiqués sur la figure 3.16(b), cela montre clairement que l'utilisation de la platitude pour résoudre ce type de problème peut apporter un gain de performances non négligeable, qui peut alors être mis à profit pour optimiser des trajectoires à bord. Cela présentera un intérêt évident lorsque nous traiterons de la replanification de trajectoires à bord en présence de défauts actionneurs dans la section 3.6.

### Seconde formulation : paramétrisation en fonction de l'énergie totale

La seconde formulation que nous avons retenu dans le cadre de cette étude est peut être plus en adéquation avec la problématique de planification de trajectoires TAEM. En effet, comme nous cherchons à dissiper l'énergie du véhicule de manière continue le long de la trajectoire, la paramétrisation directe du problème de commande optimale en fonction de l'énergie totale semble être une démarche plus naturelle, et peut être mieux adaptée. Nous allons alors montrer qu'il est possible, moyennant, comme dans la section précédente, la satisfaction d'une contrainte égalitaire de trajectoire additionnelle, de transcrire le problème de commande optimale (3.15) dans l'espace des sorties plates.

Nous allons commencer par réécrire les équations dynamiques (3.3)-(3.8) régissant le mouvement du centre de gravité du véhicule en fonction de l'énergie totale, en considérant que cinq des variables d'état peuvent s'exprimer comme des fonctions d'une sixième variable étant strictement monotone le long de la trajectoire considérée. Un choix naturel consiste alors à retenir l'énergie totale  $E$ , strictement décroissante, comme variable indépendante. Cette dernière est définie par l'équation classique

$$E = \frac{1}{2}mV^2 + mgh. \quad (3.45)$$

Maintenant, en associant à l'énergie totale la variable<sup>59</sup> normalisée croissante  $E_n$  telle que

$$E = E_0 + (E_f - E_0) E_n, \quad 0 \leq E_n \leq 1, \quad (3.46)$$

il vient

$$(\cdot)' = \frac{d(\cdot)}{dE_n} = \frac{E_f - E_0}{\dot{E}} \frac{d(\cdot)}{dt}, \quad (3.47)$$

où la dérivée de l'énergie est donnée par  $\dot{E} = -DV$ .

Maintenant, en utilisant le changement de variables (3.47), et en introduisant la contrainte de vol symétrique  $\beta = 0$  dans les équations (3.3)-(3.8), ces dernières se réécrivent

$$x' = \frac{E_0 - E_f}{D} \cos \chi \cos \gamma, \quad (3.48)$$

$$y' = \frac{E_0 - E_f}{D} \sin \chi \cos \gamma, \quad (3.49)$$

$$h' = \frac{E_0 - E_f}{D} \sin \gamma, \quad (3.50)$$

$$\gamma' = \frac{E_0 - E_f}{DV} \left( \frac{L \cos \mu}{mV} - \frac{g}{V} \cos \gamma \right), \quad (3.51)$$

$$\chi' = \frac{E_0 - E_f}{DV} \frac{L \sin \mu}{mV \cos \gamma}, \quad (3.52)$$

où les bornes  $E_0 = \frac{1}{2}mV_0^2 + mgh_0$  et  $E_f = \frac{1}{2}mV_f^2 + mgh_f$  sont déterminées en fonction des conditions initiales et finales de trajectoire considérées. De plus, comme la vitesse relative  $V$  peut être déterminée à partir de  $h$  et de  $E$ , seulement cinq équations sont nécessaires.

En choisissant à nouveau le triplet  $(x, y, h)$  comme sortie plate du modèle (3.48)-(3.52) (cf proposition 3.1), on peut alors réécrire l'état  $(x, y, h, \gamma, \chi)$  et les commandes  $(\alpha, \mu)$  du système (3.48)-(3.52), ainsi que la vitesse relative  $V$ , en fonction de l'énergie totale normalisée  $E_n$ , de la portée longitudinale  $x = z_1$ , de la portée latérale  $y = z_2$  et de l'altitude  $h = z_3$ , et de leurs dérivées. Les états du modèle se réécrivent alors sous la forme

$$V = \sqrt{2 \left( \frac{E}{m} - gz_3 \right)}, \quad (3.53)$$

$$\gamma = \arctan \left( \frac{z_3'}{\sqrt{z_1'^2 + z_2'^2}} \right), \quad (3.54)$$

$$\chi = \arctan \left( \frac{z_2'}{z_1'} \right), \quad (3.55)$$

où  $E$  est définie par l'expression (3.46), qui dépend bien de l'énergie totale normalisée  $E_n$ .

---

<sup>59</sup>on appellera dans la suite cette grandeur énergie totale normalisée par abus de langage.

Maintenant, en utilisant les équations (3.48), (3.49) et (3.50), la force de traînée  $D$  peut s'écrire

$$D = \frac{E_0 - E_f}{\sqrt{z_1'^2 + z_2'^2 + z_3'^2}}, \quad (3.56)$$

L'expression de l'angle de gîte aérodynamique est alors déterminée à partir des expressions (3.51), (3.52), et en utilisant (3.56), telle que

$$\mu = \arctan \left( \frac{mV^2 \chi' \cos \gamma}{mV^2 \gamma' + mg \cos \gamma \sqrt{z_1'^2 + z_2'^2 + z_3'^2}} \right), \quad (3.57)$$

où  $\chi'$  et  $\gamma'$  sont respectivement définis par (3.39) et (3.40), sauf que l'opération de dérivation  $(.)'$  est effectuée par rapport à l'énergie totale normalisée  $E_n$  au lieu du temps normalisé  $\tau$ .

En remarquant que le nombre de Mach  $M$  est lié à la vitesse relative par la relation simple  $M = \frac{V}{V_s}$ , en supposant que  $V_s$  est une vitesse du son moyenne constante aux altitudes de vol considérées, le coefficient de portance en configuration lisse  $C_{L_0}$  est approximé en séparant les variables  $\alpha$  et  $M$ , puis en substituant dans l'expression de  $M$  la vitesse  $V$  définie par (3.53). On obtient donc  $C_{L_0}$  sous la forme

$$C_{L_0}(\alpha, E, z_3) = (a_0 + a_1 \alpha) f_{C_{L_0}}(E, z_3). \quad (3.58)$$

Maintenant, en élevant au carré les équations (3.51) et (3.52) pour faire apparaître respectivement les termes  $L^2 \cos^2 \mu$  et  $L^2 \sin^2 \mu$ , la force de portance peut s'écrire

$$L = \sqrt{\left( \frac{mV^2 \chi' \cos \gamma}{\sqrt{z_1'^2 + z_2'^2 + z_3'^2}} \right)^2 + \left( \frac{mV^2 \gamma'}{\sqrt{z_1'^2 + z_2'^2 + z_3'^2}} + mg \cos \gamma \right)^2}. \quad (3.59)$$

Finalement, en remplaçant  $C_{L_0}(\alpha, E, z_3)$  par l'expression (3.58) dans l'équation (3.59), où  $L = \bar{q} S C_{L_0}(\alpha, E, z_3)$ , l'expression de l'angle d'incidence est donnée par

$$\alpha = \frac{L}{a_1 f_{C_{L_0}}(E, z_3) \rho S \left( \frac{E}{m} - g z_3 \right)} - \frac{a_0}{a_1}. \quad (3.60)$$

Cependant, le système (3.48)-(3.52) n'étant pas plat, il existe à nouveau une équation autonome. En effet, la force de traînée  $D$  peut être calculée soit à partir de l'équation (3.56), soit en utilisant l'expression  $D = \frac{1}{2} \rho S V^2 C_{D_0}(\alpha, M)$ . Ainsi, la contrainte égalitaire de trajectoire additionnelle, définie par

$$\Lambda_{E_n}(x, u) = D - \rho S \left( \frac{E}{m} - g z_3 \right) C_{D_0}(\alpha, E, z_3) = 0, \quad (3.61)$$

doit être satisfaite le long de la trajectoire TAEM, où la force de traînée  $D$  est définie par (3.56). De plus, le coefficient de traînée en configuration lisse  $C_{D_0}(\alpha, M)$  est approximé

par un réseau de neurones statique, puis l'expression (3.53) est substituée dans  $M = V/V_s$  pour obtenir une expression du type  $C_{D_0}(\alpha, E, z_3)$ . Notons enfin que la contrainte (3.61) peut s'exprimer en fonction des sorties plates  $z_1, z_2, z_3$  et de leurs dérivées, ainsi qu'en fonction de l'énergie totale normalisée  $E_n$ .

Finalement, les contraintes mécaniques de trajectoires, c'est-à-dire la pression dynamique  $\bar{q}$  définie par (3.12) et la contrainte de facteur de charge (3.14) peuvent également s'écrire en fonction des sorties plates, de leurs dérivées, ainsi que de l'énergie totale normalisée  $E_n$ .

Grâce aux relations d'équivalence (2.5), fournissant les expressions des états et des commandes en fonction des sorties plates, c'est à dire les équations (3.53)-(3.55) et (3.57)-(3.60), nous pouvons à présent transcrire le problème de commande optimale (3.15) dans l'espace des sorties plates. Ce dernier est alors défini par le problème de faisabilité

$$\begin{aligned}
 & \text{trouver } \bar{z}(t) & (3.62) \\
 & \text{t.q.} \\
 & \psi_x(\bar{z}(E_{n_0}), E_n) = x_0, \\
 & \psi_u(\bar{z}(E_{n_0}), E_n) = u_0, \\
 & \Lambda_{E_n}(\psi_x(\bar{z}(E_n), E_n), \psi_u(\bar{z}(E_n), E_n)) = 0, & E_n \in [E_{n_0}, E_{n_f}], \\
 & 0 \leq \Gamma(\psi_x(\bar{z}(E_n), E_n), \psi_u(\bar{z}(E_n), E_n)) \leq \Gamma_{\max}, & E_n \in [E_{n_0}, E_{n_f}], \\
 & 0 \leq \bar{q}(\psi_x(\bar{z}(E_n), E_n)) \leq \bar{q}_{\max}, & E_n \in [E_{n_0}, E_{n_f}], \\
 & u_{\min} \leq \psi_u(\bar{z}(E_n), E_n) \leq u_{\max}, & E_n \in [E_{n_0}, E_{n_f}], \\
 & \psi_x(\bar{z}(E_{n_f}), E_n) = x_f, \\
 & \psi_u(\bar{z}(E_{n_f}), E_n) = u_f,
 \end{aligned}$$

où le vecteur  $\bar{z}$  est défini par  $\bar{z} = (z_1, z_2, z_3, \dot{z}_1, \dot{z}_2, \dot{z}_3, \ddot{z}_1, \ddot{z}_2, \ddot{z}_3)$  dans l'espace des jets d'ordre infini, et où l'énergie totale normalisée  $E_n$  vérifie  $E_{n_0} = 0$  et  $E_{n_f} = 1$ . En outre, la contrainte de trajectoire  $\Lambda_{E_n}$  est définie par l'équation autonome (3.61).

Afin d'illustrer la démarche de planification de trajectoires proposée, nous reprenons l'exemple illustratif décrit dans la section précédente. Nous supposons que nous sommes en présence de conditions initiales non dispersées, définies dans la troisième colonne du tableau 3.3. De plus, les conditions finales de trajectoire au point ALI correspondent à nouveau à celles indiquées dans le tableau 3.3, ainsi que les bornes sur les consignes de guidage apparaissant dans le tableau 3.4.

Afin de résoudre le problème de commande optimale (3.62), il est nécessaire de le transformer en un problème de programmation non linéaire. Nous avons alors paramétré les sorties plates  $z_1(t)$ ,  $z_2(t)$  et  $z_3(t)$  par des B-splines d'ordre 6, de multiplicité 4 et comportant 6 intervalles. La variable monotonement croissante  $E_n$  associée à l'énergie totale normalisée est paramétrée par une B-spline d'ordre 2, de multiplicité 1 et comportant 1 intervalle. En outre, l'OCP (3.62) est discrétisé en 24 points de collocation, uniformément répartis entre  $E_{n_0}$  et  $E_{n_f}$ , où les contraintes de trajectoires sont satisfaites. Nous avons également utilisé de nouveau le solveur NPSOL [44].

**Remarque 3.18.** Comme ce fut le cas pour la variable temporelle dans la section pré-

cédente, les sorties plates fonction de l'énergie totale sont normalisées sous la forme  $z_i = z_{i_0} + (z_{i_f} - z_{i_0})z_{i_n}$  avec  $0 \leq z_{i_n} \leq 1$  pour  $i = 1, \dots, 3$ , afin d'améliorer le conditionnement numérique du problème, en particulier le conditionnement des matrices jacobiennes et hessiennes calculées par le solveur NLP.

La figure 3.20 illustre la trajectoire TAEM obtenue en 0.19 s sous l'environnement Matlab. Le bandeau de couleurs sur la partie droite représente le niveau du facteur de charge le long de la trajectoire TAEM.

**Remarque 3.19.** A la vue de la figure 3.20, le véhicule semble s'enrouler autour du HAC, bien qu'il ne soit pas pris en compte dans notre approche. Aussi, comparé à d'autres techniques [93], la méthode proposée ne requiert pas de stratégie de dimensionnement du HAC, ce dernier étant représenté sur la figure 3.20 (ainsi que dans la suite du manuscrit) uniquement à titre illustratif.

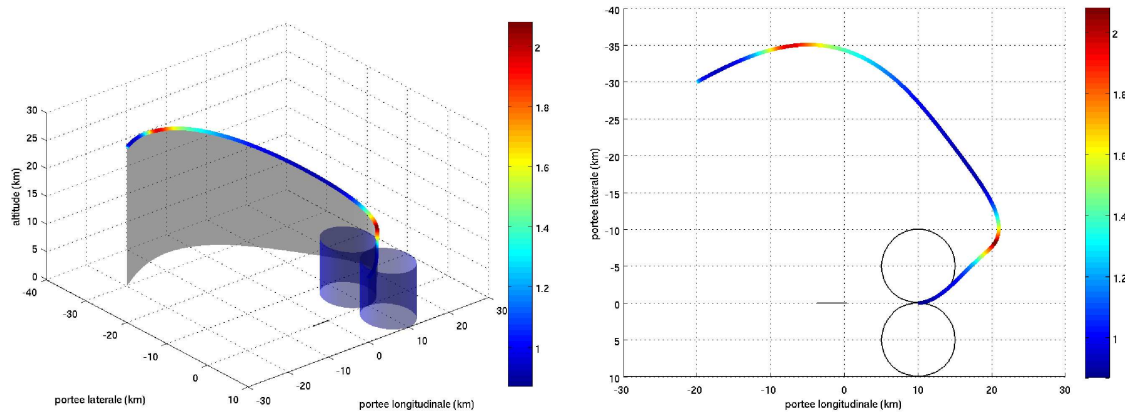


FIG. 3.20 – trajectoire TAEM nominale.

Les consignes de guidage  $\alpha$  et  $\mu$  associées sont représentées sur les figures 3.21(a) et 3.21(b) en fonction de l'énergie totale.

De plus, la contrainte non linéaire de trajectoire (3.61) est bien vérifiée pour l'ensemble des points de collocation, comme indiqué sur la figure 3.22.

Finalement, les contraintes de pression dynamique et de facteur de charge le long de la trajectoire TAEM sont représentées sur les figures 3.23(a) et 3.23(b) respectivement.

### 3.3.3 Planification de trajectoires A&L par platitude

Dans cette section, nous allons utiliser la propriété de platitude du modèle longitudinal du véhicule (3.16)-(3.19) afin de transcrire le problème de commande optimale (3.20) associé à la phase A&L dans l'espace des sorties plates.

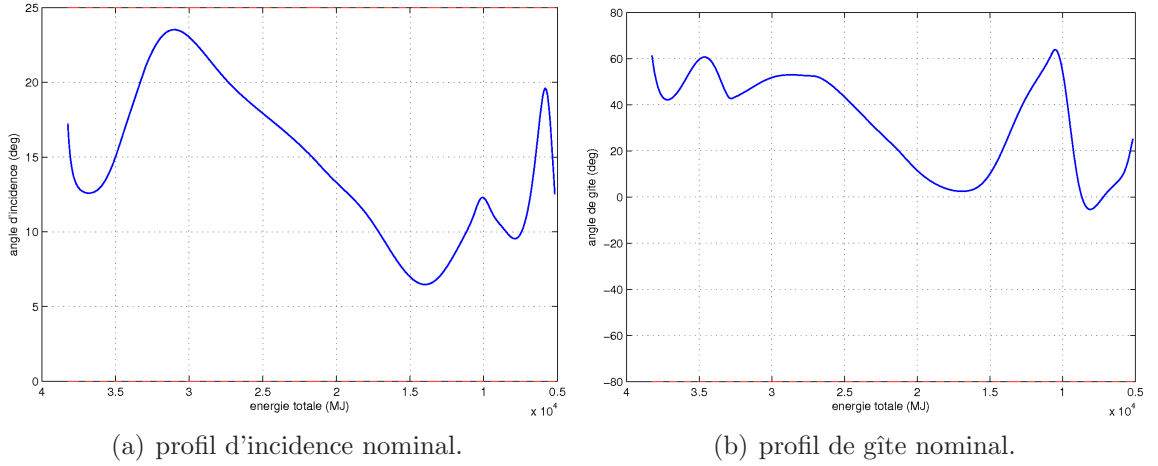


FIG. 3.21 – profils des consignes de guidage de référence.

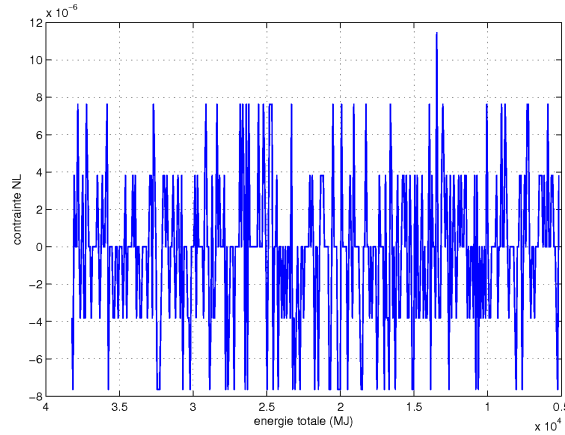


FIG. 3.22 – contrainte égalitaire de trajectoire ( $N$ ).

En ce qui concerne la phase d'atterrissage automatique qui, rappelons-le, considère uniquement le mouvement longitudinal du véhicule, nous allons commencer par paramétrer les équations (3.16)-(3.19) en fonction de la portée longitudinale restante par rapport au point de toucher des roues sur la piste. En posant  $x = x_0 + (x_f - x_0)x_n$ , où  $x_n$  désigne la portée longitudinale normalisée, on peut alors définir le changement de variable suivant :

$$(\cdot)' = \frac{d(\cdot)}{dx_n} = \frac{x_f - x_0}{\dot{x}} \frac{d(\cdot)}{dt}, \quad (3.63)$$

où  $\dot{x}$  est définie par l'équation (3.16).

Les équations longitudinales (3.16)-(3.19), régissant la dynamique du véhicule lors de la phase A&L, peuvent alors se réécrire sous la forme d'un système de trois équations



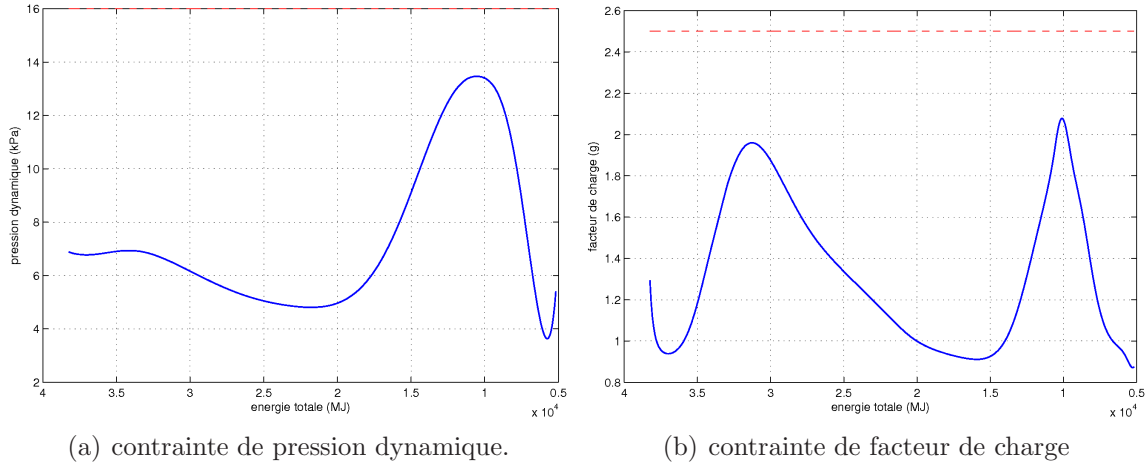


FIG. 3.23 – contraintes mécaniques le long de la trajectoire TAEM de référence.

différentielles, tel que

$$h' = (x_f - x_0) \tan \gamma, \quad (3.64)$$

$$V' = \frac{x_f - x_0}{V \cos \gamma} \left( -\frac{D}{m} - g \sin \gamma \right), \quad (3.65)$$

$$\gamma' = \frac{x_f - x_0}{V \cos \gamma} \left( \frac{L}{mV} - \frac{g}{V} \cos \gamma \right), \quad (3.66)$$

$$(3.67)$$

où les valeurs de  $x_0$  et de  $x_f$  sont déterminées en fonction des conditions initiales et finales de trajectoire considérées.

Le modèle (3.64)-(3.66) comporte trois états et une seule commande, l'angle d'incidence  $\alpha$ . Il n'est donc pas plat. Cependant, l'état et la consigne d'incidence peuvent se réécrire en fonction de l'altitude  $h$  et de la vitesse relative  $V$ , ainsi que de leurs dérivées, moyennant la satisfaction d'une équation autonome le long de la trajectoire A&L. En posant  $z_1 = h$  et  $z_2 = V$ , la pente de trajectoire  $\gamma$  peut alors s'écrire sous la forme

$$\gamma = \arctan \left( \frac{z_3'}{x_f - x_0} \right). \quad (3.68)$$

Maintenant, la consigne d'incidence peut être déterminée à partir de l'équation (3.66), telle que

$$\alpha = \frac{2m \cos \gamma \left( \frac{\gamma' z_2^2}{x_f - x_0} + g \right)}{a_1 f_{C_{L_0}}(z_2) \rho S z_2^2} - \frac{a_0}{a_1}, \quad (3.69)$$

où

$$\rho = \rho_0 \exp(-z_1/H_0), \quad (3.70)$$

et

$$\gamma' = \frac{(x_f - x_0)z_1''}{(x_f - x_0)^2 + z_1'} \quad (3.71)$$

De plus, en notant que  $z_2 = MV_s$ , nous avons modélisé le coefficient de portance en configuration lisse  $C_{L_0}(\alpha, M)$  sous la forme  $\sigma_1 u_1(\alpha) v_1(M)$ , tel que

$$C_{L_0}(\alpha, z_2) = (a_0 + a_1 \alpha) f_{C_{L_0}}(z_2), \quad (3.72)$$

en utilisant l'approche par séparation de variables et réseaux de neurones décrite dans la section 3.3.1.

**Remarque 3.20.** Pour les nombres de Mach considérés (inférieurs à 0.5), on peut facilement vérifier que l'effet du nombre de Mach sur le coefficient de portance  $C_{L_0}$  est négligeable. Ainsi, en fixant une valeur de  $M$  moyenne, il est possible d'approximer  $C_{L_0}$  en fonction de l'incidence uniquement, et ainsi de poser  $f_{C_{L_0}}(z_2) = 1$  dans l'expression (3.72). C'est précisément ce que nous avons choisi de faire dans la suite.

Comme le système (3.64)-(3.66) est sous-actionné, et donc non plat, il est alors nécessaire de satisfaire une contrainte égalitaire de trajectoire supplémentaire, associée à l'observable autonome du système. Ainsi, l'équation autonome est définie par (3.65), qu'on peut alors réécrire sous la forme

$$\Lambda_{x_n}(x, u) = V' - \frac{x_f - x_0}{V \cos \gamma} \left( -\frac{\rho S V^2 C_{D_0}(\alpha, M)}{2m} - g \sin \gamma \right), \quad (3.73)$$

où le coefficient de traînée en configuration lisse  $C_{D_0}(\alpha, M)$  est approximé par une fonction analytique dépendant uniquement de l'angle d'incidence pour un nombre de Mach moyen prédéfini. Notons que  $\Lambda_{x_n}$  peut s'exprimer exclusivement en fonction de  $z_1$ ,  $z_2$ , et de leurs dérivées successives.

Finalement, la contrainte de pression dynamique  $\bar{q}$  définie par (3.12) et la contrainte de facteur de charge donnée par (3.14) peuvent également s'écrire en fonction de  $z_1$  et  $z_2$ , de leurs dérivées, ainsi que de la portée longitudinale normalisée  $x_n$ .

Nous sommes maintenant en mesure de réécrire le problème de commande optimale (3.20) dans l'espace des sorties plates, en utilisant les relations d'équivalence (2.5), fournissant les expressions des états et des commandes en fonction des sorties plates. Ainsi, en utilisant les expressions (3.68) et (3.69), le problème de commande optimale (3.20) peut se

réécrire sous la forme

$$\begin{aligned}
 & \text{trouver } \bar{z}(t) & (3.74) \\
 & \text{t.q.} \\
 & \psi_x(\bar{z}(x_{n_0}), x_n) = x_0, \\
 & \psi_\alpha(\bar{z}(x_{n_0}), x_n) = \alpha_0, \\
 & \Lambda_{x_n}(\psi_x(\bar{z}(x_n), x_n), \psi_u(\bar{z}(x_n), x_n)) = 0, & x_n \in [x_{n_0}, x_{n_f}], \\
 & 0 \leq \Gamma(\psi_x(\bar{z}(x_n), x_n), \psi_u(\bar{z}(x_n), x_n)) \leq \Gamma_{\max}, & x_n \in [x_{n_0}, x_{n_f}], \\
 & 0 \leq \bar{q}(\psi_x(\bar{z}(x_n), x_n)) \leq \bar{q}_{\max}, & x_n \in [x_{n_0}, x_{n_f}], \\
 & \alpha_{\min} \leq \psi_\alpha(\bar{z}(x_n), x_n) \leq \alpha_{\max}, & x_n \in [x_{n_0}, x_{n_f}], \\
 & \psi_x(\bar{z}(x_{n_f}), x_n) = x_f, \\
 & \psi_\alpha(\bar{z}(x_{n_f}), x_n) = \alpha_f,
 \end{aligned}$$

où le vecteur  $\bar{z}$  est défini par  $\bar{z} = (z_1, \dot{z}_1, \ddot{z}_1, z_2, \dot{z}_2)$  dans l'espace des jets d'ordre infini. De plus, par définition, la portée longitudinale normalisée  $x_n$  vérifie  $x_{n_0} = 0$  et par  $x_{n_f} = 1$ .

**Remarque 3.21.** Nous n'avons pas pris en compte une fonctionnelle de coût particulière dans le problème de planification de trajectoires A&L défini par (3.74). De ce fait, nous nous limitons à un problème de faisabilité dans le cadre de cette étude.

Afin d'illustrer la démarche de planification de trajectoires A&L, nous avons considéré que les conditions initiales au point ALI correspondent aux conditions finales de la trajectoire TAEM, définies dans le tableau 3.3. Les conditions cinématiques au point de toucher des roues sur la piste sont données dans le tableau 3.5. De plus, les contraintes associées à l'angle d'incidence sont définies dans le tableau 3.4. Finalement, les valeurs maximales des contraintes de facteur de charge et de pression dynamique sont fixées respectivement à 2.5 g et 16 kPa, comme pour la phase TAEM.

Le problème de commande optimale (3.74) est ensuite transformé en un problème NLP, en paramétrant les variables d'optimisation par des B-splines, puis en discrétisant les équations en un certain nombre de points équidistants. Ainsi, nous avons choisi des B-splines d'ordre 6, de multiplicité 3 et comportant 8 intervalles afin de paramétrer les sorties plates  $z_1$  et  $z_2$ . De plus, la portée longitudinale normalisée  $x_n$  est décrite par une B-spline d'ordre 2, de multiplicité 1 et comportant 1 seul segment. En outre, les contraintes de trajectoire sont vérifiées en 32 points de collocation uniformément répartis entre  $x_{n_0}$  et  $x_{n_f}$ . Le problème de programmation non linéaire résultant a ensuite été résolu via le solveur NPSOL, en 20 ms environ (temps CPU) sur un ordinateur personnel équipé d'un processeur Intel Core 2 Duo cadencé à 2 GHz.

Même s'il existe une solution admissible au problème précédent, le résultat de cette première optimisation engendre un profil d'incidence relativement proche de la valeur d'incidence maximale admissible  $\alpha_{\max} = 25^\circ$  (voir figure 3.24).

Afin de relâcher les contraintes associées au problème de planification de trajectoires A&L, nous avons décidé d'augmenter la longueur de la trajectoire devant être parcourue jusqu'au point de toucher, en modifiant simplement la portée longitudinale initiale pour

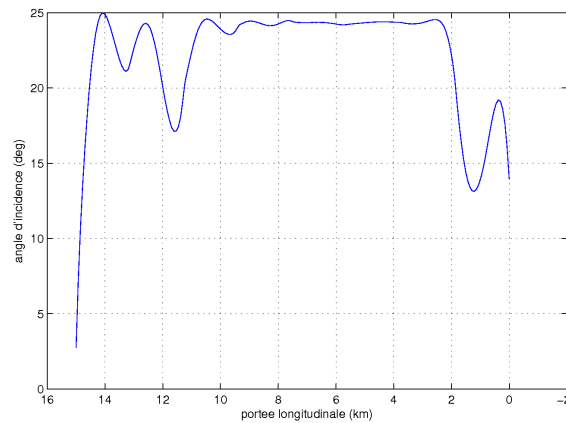


FIG. 3.24 – profil nominal d'incidence pour la condition initiale  $x_0 = -10$  km.

$x_0 = -20$  km. La figure 3.25 illustre la trajectoire d'atterrissage automatique obtenue en 20 ms via le solveur NPSOL. Le bandeau de couleurs sur la partie droite représente le niveau du facteur de charge le long de la trajectoire TAEM.

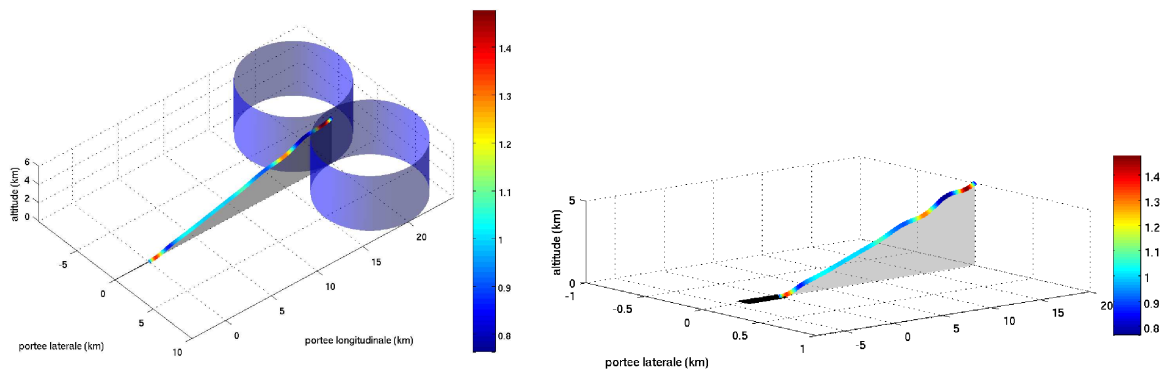


FIG. 3.25 – trajectoire A&L nominale.

Les profils des états du système correspondant à cette trajectoire A&L sont représentés sur les figures 3.26(a), 3.26(b) et 3.26(c) en fonction de la portée longitudinale restante.

La consigne d'incidence peut ensuite être calculée le long de la trajectoire A&L nominale en utilisant l'expression (3.69). Le profil correspondant est représenté sur la figure 3.27(a). Contrairement au profil obtenu en considérant la condition initiale  $x_0 = -10$  km (cf figure 3.24), les valeurs d'incidence obtenues ici sont mieux centrées dans le corridor admissible  $0^\circ \leq \alpha \leq 25^\circ$ .

De plus, la contrainte égalitaire de trajectoire (3.73) est bien vérifiée pour l'ensemble des points de collocation, comme indiqué sur la figure 3.27(b).

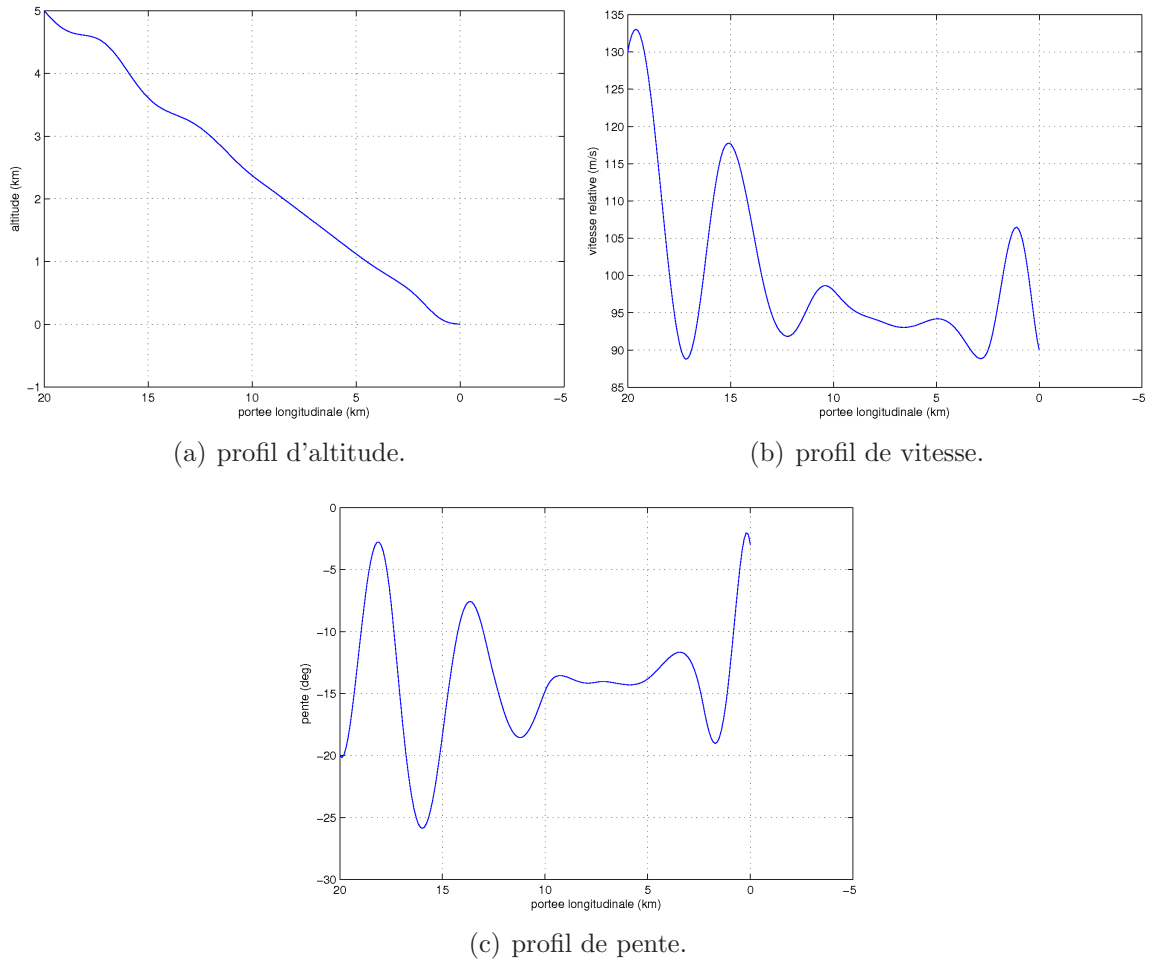


FIG. 3.26 – états de référence sur la trajectoire A&L.

Finalement, les contraintes de pression dynamique et de facteur de charge le long de la trajectoire d'atterrissage sont représentées sur les figures 3.28(a) et 3.28(b).

### 3.4 Planification de trajectoires de rentrée en présence de perturbations aérologiques

L'objectif de cette section est la prise en compte de perturbations aérologiques dans l'algorithme de planification de trajectoires. Comme nous allons le voir, l'effet de ces perturbations atmosphériques peut impacter de manière non négligeable le profil des trajectoires générées. Il est alors important de tenir compte des effets du vent sur la dynamique du véhicule, où l'atmosphère possède un mouvement non uniforme et non stationnaire. Ce type de perturbation a été étudié par [181, 182] dans le cadre de la planification de trajectoires d'atterrissage automatique en vol. L'effet du vent est alors considéré comme

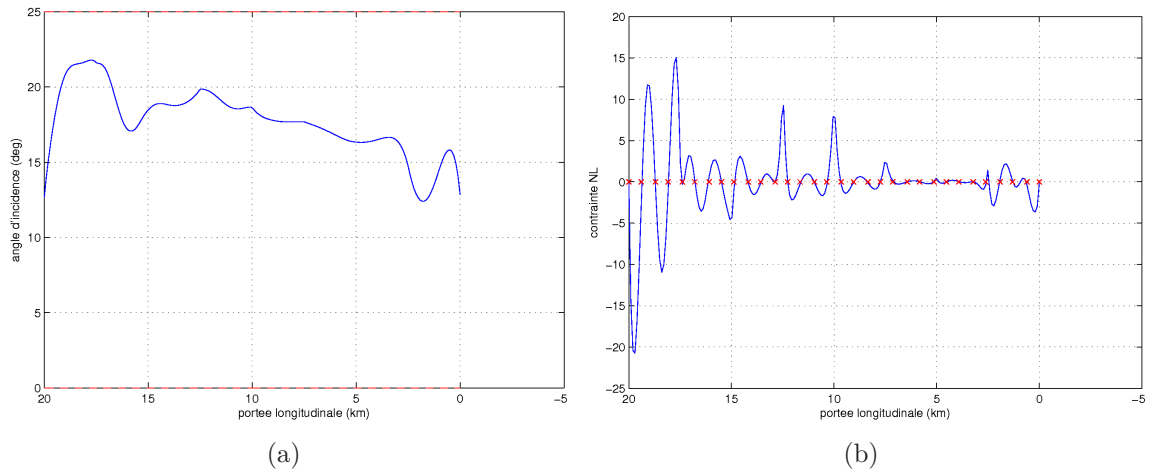


FIG. 3.27 – (a) profil nominal d'incidence pour la condition initiale  $x_0 = -20$  km (b) contrainte égalitaire de trajectoire ( $m/s$ ). Les points de collocation sont représentés par des croix rouges.

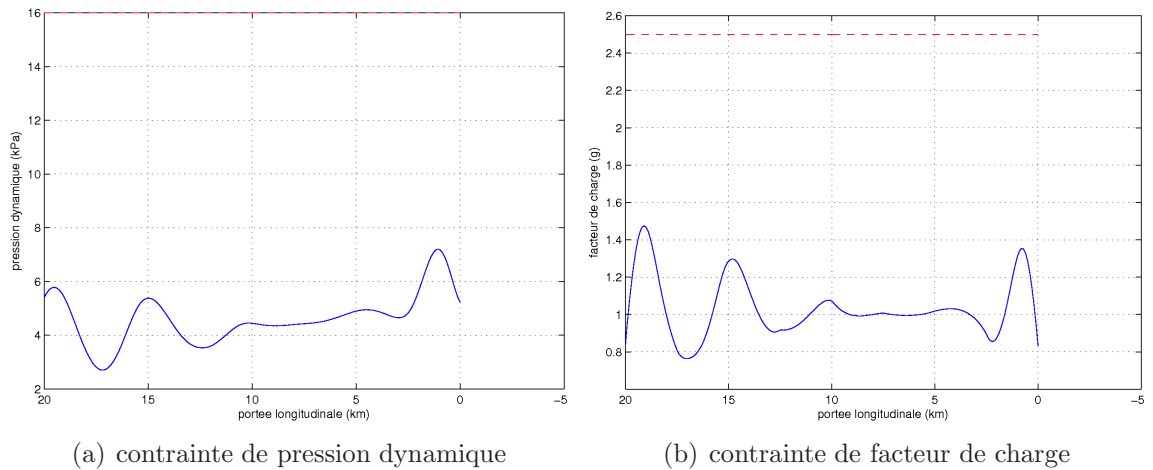


FIG. 3.28 – contraintes mécaniques le long de la trajectoire A&L de référence.

une perturbation sur le modèle du véhicule qu'un correcteur adaptatif permet alors d'éliminer. Plutôt que de s'efforcer d'éliminer l'effet du vent en robustifiant le régulateur de suivi de trajectoires, il peut être intéressant de l'incorporer directement dans le processus de planification de trajectoires, afin de limiter l'action du régulateur de poursuite de trajectoire. Une stratégie de ce type a été proposée dans l'article [183] pour la phase d'atterrissage d'un avion de transport civil, où l'auteur propose de déterminer des trajectoires optimales et sous-optimales dans un plan vertical en présence de cisaillements de vent.

Nous supposons dans la suite que, pour la durée de la trajectoire considérée, la distribution spatiale moyenne des vitesses aérologiques est stationnaire et peut être estimée

par le segment sol.

### 3.4.1 Champs de cisaillement de vent considérés

Lors d'une variation locale du vecteur vent, il se produit un phénomène de cisaillement. Les variations aérologiques en vitesse et en direction sont estimées dans les directions horizontales et verticales. Un cisaillement de vent horizontal implique une variation du vent le long de la portée longitudinale et de la portée latérale de la trajectoire. Dans le cas d'un cisaillement de vent vertical, la vitesse et la direction du vent varient principalement en fonction de l'altitude. Les cisaillements de vent sont créés par le mouvement des masses d'air entre elles ou par rapport à la surface terrestre<sup>60</sup>. Les orages, les systèmes frontaux, et la couche limite terrestre produisent chacun des cisaillements de vent qui, parfois, peuvent se révéler être dangereux pour les avions volant à basse altitude. Afin d'analyser l'influence des cisaillements de vent sur le comportement dynamique du véhicule, les caractéristiques de ces cisaillements de vent doivent être connues. Il n'existe pas de formulation mathématique simple qui permette de caractériser les cisaillements de vents produits par les orages et les systèmes frontaux. En revanche, les effets du cisaillement de vent produit par les couches limites de l'atmosphère peuvent être analysés mathématiquement. Ainsi, ces effets peuvent être modélisés en considérant un champ de vitesse réparti sur les trois axes du repère cartésien local attaché à la piste. Le vecteur vent peut alors avoir une composante longitudinale, latérale, et verticale. Nous considérons pour cela un champ général de cisaillement de vent  $(\tau_x, \tau_y, \tau_h)$ , défini par

$$\begin{cases} \tau_x(x, y, h) = K_{x_1} x^{\sigma_1} y^{\nu_1} h^{\lambda_1} + K_{x_2}, \\ \tau_y(x, y, h) = K_{y_1} x^{\sigma_2} y^{\nu_2} h^{\lambda_2} + K_{y_2}, \\ \tau_h(x, y, h) = K_{h_1} x^{\sigma_3} y^{\nu_3} h^{\lambda_3} + K_{h_2}. \end{cases} \quad (3.75)$$

où les coefficients  $(K_{x_1}, K_{y_1}, K_{h_1})$  représentent les variations de la norme du module de la vitesse vent,  $(K_{x_2}, K_{y_2}, K_{h_2})$  sont des biais constants, et où  $(\sigma_i, \nu_i, \lambda_i)$ ,  $i = 1, \dots, 3$ , sont des puissances non négatives. Plus précisément, le modèle mathématique (3.75) représente une extension en trois dimensions du modèle de cisaillement de vent vertical proposé dans [184].

### 3.4.2 Intégration des perturbations aérologiques dans le processus de planification de trajectoires

Nous allons maintenant chercher à intégrer le modèle de vent défini par (3.75) dans l'algorithme de planification de trajectoires TAEM à bord. pour cela, nous considérons à nouveau le modèle du véhicule à trois degrés de liberté (3.48)-(3.52), paramétré en fonction de l'énergie totale normalisée, que nous avons introduit dans la section 3.3.2, et dans lequel les contributions du champ de cisaillement de vent sur les trois axes ont été

---

<sup>60</sup>ce phénomène est bien connu en très basse altitude lors de l'atterrissage

prises en compte. On obtient alors

$$x' = \frac{E_0 - E_f}{D} \cos \chi \cos \gamma + \tau_x(x, y, h), \quad (3.76)$$

$$y' = \frac{E_0 - E_f}{D} \sin \chi \cos \gamma + \tau_y(x, y, h), \quad (3.77)$$

$$h' = \frac{E_0 - E_f}{D} \sin \gamma + \tau_h(x, y, h), \quad (3.78)$$

$$\gamma' = \frac{E_0 - E_f}{DV} \left( \frac{L \cos \mu}{mV} - \frac{g}{V} \cos \gamma \right), \quad (3.79)$$

$$\chi' = \frac{E_0 - E_f}{DV} \frac{L \sin \mu}{mV \cos \gamma}, \quad (3.80)$$

dans lequel apparaissent les différents termes du modèle (3.75). De plus, afin de démontrer la faisabilité de notre approche, nous considérons que les termes  $\tau_x$ ,  $\tau_y$  et  $\tau_h$  s'expriment en unités de longueur.

**Remarque 3.22.** Nous aurions également pu remplacer les termes  $\tau_x$ ,  $\tau_y$  et  $\tau_h$  de la partie droite du modèle (3.76)-(3.80) par les expressions suivantes :

$$\tilde{\tau}_x(x, y, h) = \frac{E_0 - E_f}{DV} \tau_x(x, y, h),$$

$$\tilde{\tau}_y(x, y, h) = \frac{E_0 - E_f}{DV} \tau_y(x, y, h),$$

$$\tilde{\tau}_h(x, y, h) = \frac{E_0 - E_f}{DV} \tau_h(x, y, h),$$

où les termes  $\tilde{\tau}_x$ ,  $\tilde{\tau}_y$  et  $\tilde{\tau}_h$  sont alors classiquement définis en m/s.

En utilisant à nouveau les paramètres de position du centre de gravité du véhicule  $z_1 = x$ ,  $z_2 = y$  et  $z_3 = h$ , il est possible d'exprimer l'état et les commandes du modèle (3.76)-(3.80) en fonction de ces dernières et d'un nombre fini de leurs dérivées énergétiques, ainsi que de l'énergie totale normalisée  $E_n$ . L'état du système peut alors s'écrire

$$V = \sqrt{2 \left( \frac{E}{m} - g z_3 \right)}, \quad (3.81)$$

$$\gamma = \arctan \left( \frac{z_3' - \tau_h}{\sqrt{(z_1' - \tau_x)^2 + (z_2' - \tau_y)^2}} \right), \quad (3.82)$$

$$\chi = \arctan \left( \frac{z_2' - \tau_y}{z_1' - \tau_x} \right). \quad (3.83)$$

où  $\tau_x$ ,  $\tau_y$  et  $\tau_h$  s'expriment simplement en fonction de  $z_1$ ,  $z_2$  et  $z_3$ . De plus,  $E$  est définie par l'équation (3.46), qui est fonction de l'énergie totale normalisée  $E_n$ .



Il est alors possible, à partir des équations (3.76), (3.77) et (3.78), de calculer la force de traînée  $D$  sous la forme

$$D = \frac{E_0 - E_f}{\sqrt{(z'_1 - \tau_x)^2 + (z'_2 - \tau_y)^2 + (z'_3 - \tau_h)^2}}. \quad (3.84)$$

La consigne d'angle de gîte aérodynamique peut ensuite être déterminée en utilisant les expressions (3.79), (3.80) et (3.84), telle que

$$\mu = \arctan \left( \frac{mV^2 \chi' \cos \gamma}{mV^2 \gamma' + mg \cos \gamma \sqrt{(z'_1 - \tau_x)^2 + (z'_2 - \tau_y)^2 + (z'_3 - \tau_h)^2}} \right), \quad (3.85)$$

où les dérivées  $\chi'$  et  $\gamma'$  sont définies par

$$\chi' = \frac{(z''_2 - \tau'_y)(z'_1 - \tau_x) - (z'_2 - \tau_y)(z''_1 - \tau'_x)}{(z'_1 - \tau_x)^2 + (z'_2 - \tau_y)^2} \quad (3.86)$$

et

$$\gamma' = \frac{(z''_3 - \tau'_h) [(z'_1 - \tau_x)^2 + (z'_2 - \tau_y)^2] - (z'_3 - \tau_h) [(z'_1 - \tau_x)(z''_1 - \tau'_x) + (z'_2 - \tau_y)(z''_2 - \tau'_y)]}{((z'_1 - \tau_x)^2 + (z'_2 - \tau_y)^2 + (z'_3 - \tau_h)^2) \sqrt{(z'_1 - \tau_x)^2 + (z'_2 - \tau_y)^2}}. \quad (3.87)$$

De plus, les dérivées des composantes aérologiques  $\tau'_x$ ,  $\tau'_y$  et  $\tau'_h$  en fonction de l'énergie totale normalisée sont données par

$$\begin{cases} \tau'_x = K_{x1}(\sigma_1 z_1^{(\sigma_1-1)} z'_1 z_2^{\nu_1} z_3^{\lambda_1} + \nu_1 z_2^{(\nu_1-1)} z'_2 z_1^{\sigma_1} z_3^{\lambda_1} + \lambda_1 z_3^{(\lambda_1-1)} z'_3 z_1^{\sigma_1} z_2^{\nu_1}), \\ \tau'_y = K_{y1}(\sigma_2 z_1^{(\sigma_2-1)} z'_1 z_2^{\nu_2} z_3^{\lambda_2} + \nu_2 z_2^{(\nu_2-1)} z'_2 z_1^{\sigma_2} z_3^{\lambda_2} + \lambda_2 z_3^{(\lambda_2-1)} z'_3 z_1^{\sigma_2} z_2^{\nu_2}), \\ \tau'_h = K_{h1}(\sigma_3 z_1^{(\sigma_3-1)} z'_1 z_2^{\nu_3} z_3^{\lambda_3} + \nu_3 z_2^{(\nu_3-1)} z'_2 z_1^{\sigma_3} z_3^{\lambda_3} + \lambda_3 z_3^{(\lambda_3-1)} z'_3 z_1^{\sigma_3} z_2^{\nu_3}). \end{cases}$$

Maintenant, afin de calculer l'expression analytique de l'angle d'incidence  $\alpha$ , nous modélisons à nouveau le coefficient de portance en configuration lisse  $C_{L_0}(\alpha, M)$  grâce à l'approche par séparation de variables présentée dans la section 3.3.1. On obtient alors :

$$C_{L_0}(\alpha, M) = (a_0 + a_1 \alpha) f_{C_{L_0}}(M), \quad (3.88)$$

Finalement, en posant  $L = \bar{q} S C_{L_0}(\alpha, M)$ , où  $C_{L_0}(\alpha, M)$  est défini par (3.88), il vient

$$\alpha = \frac{L}{a_1 f_{C_{L_0}}(E, z_3) \rho S \left( \frac{E}{m} - g z_3 \right)} - \frac{a_0}{a_1}, \quad (3.89)$$

où  $L$  est extrait de l'équation (3.80) sous la forme

$$L = \frac{m D V^2 \chi' \cos \gamma}{(E_0 - E_f) \sin \mu}.$$

A nouveau, le système (3.76)-(3.80) n'est pas plat du fait de la contrainte de vol symétrique imposée. Il est alors nécessaire de satisfaire la contrainte égalitaire (3.61) le long

de la trajectoire planifiée.

En tenant compte des perturbations aérologiques dans les équations précédentes, nous obtenons alors un nouveau problème de commande optimale dans l'espace des sorties plates, dans lequel les gains  $(K_{x_1}, K_{y_1}, K_{h_1})$ , les biais  $(K_{x_2}, K_{y_2}, K_{h_2})$  et les termes de puissance  $(\sigma_i, \nu_i, \lambda_i)$ ,  $i = 1, \dots, 3$  du modèle aérologique (3.75) représentent des paramètres exogènes ajustables, regroupés dans un vecteur  $\Upsilon$  tel que

$$\Upsilon = (K_{x_1}, K_{y_1}, K_{h_1}, K_{x_2}, K_{y_2}, K_{h_2}, \sigma_1, \sigma_2, \sigma_3, \nu_1, \nu_2, \nu_3, \lambda_1, \lambda_2, \lambda_3).$$

Le problème de commande optimale s'écrit alors

$$\text{trouver } \bar{z}(t) \tag{3.90}$$

t.q.

$$\begin{aligned} & \psi_x(\bar{z}(E_{n_0}), E_n, \Upsilon) = x_0, \\ & \psi_u(\bar{z}(E_{n_0}), E_n, \Upsilon) = u_0, \\ & \Lambda_{E_n}(\psi_x(\bar{z}(E_n), \psi_u(\bar{z}(E_n), E_n, \Upsilon)) = 0, & E_n \in [E_{n_0}, E_{n_f}], \\ 0 \leq & \Gamma(\psi_x(\bar{z}(E_n), E_n), \psi_u(\bar{z}(E_n), E_n, \Upsilon)) \leq \Gamma_{\max}, & E_n \in [E_{n_0}, E_{n_f}], \\ 0 \leq & \bar{q}(\psi_x(\bar{z}(E_n), E_n, \Upsilon)) \leq \bar{q}_{\max}, & E_n \in [E_{n_0}, E_{n_f}], \\ u_{\min} \leq & \psi_u(\bar{z}(E_n), E_n, \Upsilon) \leq u_{\max}, & E_n \in [E_{n_0}, E_{n_f}], \\ & \psi_x(\bar{z}(E_{n_f}), E_n, \Upsilon) = x_f, \\ & \psi_u(\bar{z}(E_{n_f}), E_n, \Upsilon) = u_f, \end{aligned}$$

où le vecteur  $\bar{z}$  est défini, comme dans la section 3.3.2, par  $\bar{z} = (z_1, z_2, z_3, \dot{z}_1, \dot{z}_2, \dot{z}_3, \ddot{z}_1, \ddot{z}_2, \ddot{z}_3)$  dans l'espace des jets d'ordre infini (voir la section 3.3.2 pour plus de détails). Le vecteur de paramètres  $\Upsilon$  permet alors de spécifier des conditions de cisaillement de vent différentes en fonction de modèles prédéfinis.

Nous avons appliqué la démarche précédente à l'optimisation de trajectoires TAEM, en conditions nominales puis en présence de perturbations aérologiques. Nous choisis les mêmes paramètres de B-splines que dans la section 3.3.2, ainsi que le même nombre de points de collocation. Les conditions initiales du problème de planification de trajectoires sont définies dans le tableau 3.3, et les conditions cinématiques finales au point ALI dans le tableau 3.3. De plus, les bornes sur les consignes de guidage sont indiquées dans le tableau 3.4.

Les perturbations aérologiques ont été choisies de manière arbitraire, de sorte que le champ de vecteurs de cisaillement de vent soit non uniforme dans le repère  $(x, y, z)$ . Ainsi, nous avons choisi le vecteur de paramètres  $\Upsilon$  défini simplement par

$$\Upsilon = (0.08, 0.08, 0, 0, 0, -1000, 1, 0, 0, 0, 1, 0, 0, 0, 0),$$

auquel correspondent les équations

$$\begin{cases} \tau_x(x, y, h) = 0.08x, \\ \tau_y(x, y, h) = 0.08y, \\ \tau_h(x, y, h) = -1000. \end{cases}$$

Les trajectoires en situation nominale et en présence de perturbations aérologiques ont été obtenues avec des temps CPU respectifs de 0.118 s et 0.2 s. La figure 3.29 illustre les trajectoires obtenues. Le bandeau coloré sur la droite de la figure représente la norme du module des vecteurs vitesse tracés dans le repère  $(x, y, z)$ , en m/s. Clairement, l'effet des cisaillements de vent engendre des écarts de portée longitudinale, de portée latérale et d'altitude non négligeables, comparé à une trajectoire qui ne tiendrait pas compte de ces perturbations (trajectoire non dispersée sur la figure 3.29).

La figure 3.30 présente les mêmes résultats, moyennant la représentation utilisée afin d'illustrer le champ de vecteurs de cisaillement de vent. Ainsi, des lignes de champ de couleur blanche sont représentées sur un plan horizontal au niveau du sol. De plus, le quadrillage coloré indique la norme du module des vecteurs vitesse associés à la perturbation aérologique.

Il est également possible de calculer les profils des vitesses de cisaillement de vent sur les axes longitudinal, latéral et vertical, grâce aux formules suivantes :

$$\begin{aligned} V_{w_x} &= \frac{D^*V^*}{E_0 - E_f} \tau_x(z_1^*, z_2^*, z_3^*), \\ V_{w_y} &= \frac{D^*V^*}{E_0 - E_f} \tau_y(z_1^*, z_2^*, z_3^*), \\ V_{w_z} &= \frac{D^*V^*}{E_0 - E_f} \tau_h(z_1^*, z_2^*, z_3^*), \end{aligned}$$

où  $(.)^*$  représente un profil optimal calculé le long de la trajectoire TAEM. Les profils de vitesse obtenus sont représentés sur les figures 3.31(a), 3.31(b) et 3.31(c).

**Remarque 3.23.** En utilisant l'approche décrite ici, il n'est pas possible de fixer des profils de cisaillement de vent dans le domaine des vitesses<sup>61</sup>. Cependant, cela montre clairement le potentiel de la méthode proposée pour tenir compte de profils aérologiques non uniformes et non stationnaires.

Les figures 3.32(a) et 3.32(b) représentent les consignes de guidage  $\alpha$  et  $\mu$  en situation nominale et en présence d'une dispersion aérologique. Les profils d'incidence et de gîte obtenus respectent bien les contraintes d'amplitude minimale, maximale et de vitesse angulaire définies dans le tableau 3.4.

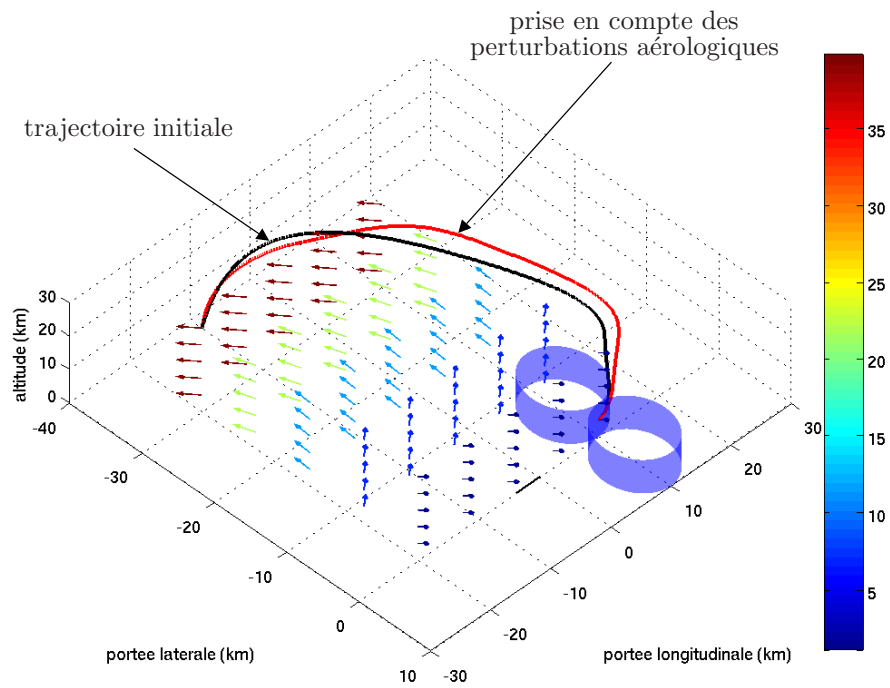
Le profil de la contrainte égalitaire de trajectoire définie par (3.61), et associée à l'observable autonome du système (3.76)-(3.80), est représenté sur la figure 3.33. En situation nominale et perturbée, la contrainte est bien satisfaite aux points de collocation.

Finalement, les figures 3.34(a) et 3.34(b) présentent les profils des contraintes de pression dynamique et de facteur de charge le long de la trajectoire TAEM. Clairement, les valeurs maximales obtenues n'excèdent pas celles définies dans le tableau 3.3.

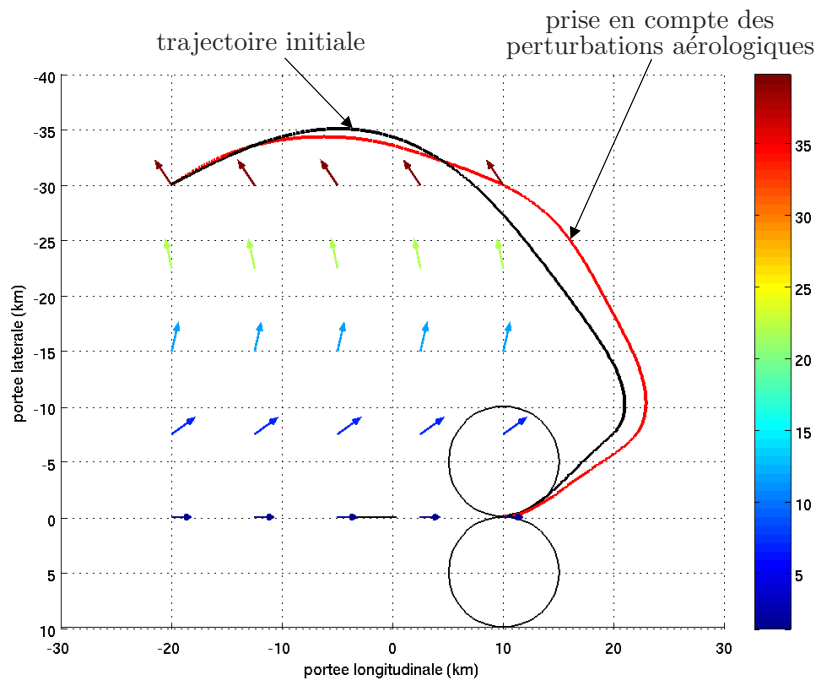
---

<sup>61</sup>le problème ne se pose plus si l'on considère le modèle à trois degrés de liberté du véhicule paramétré en fonction du temps normalisé (voir section 3.3.2).

3.4. Planification de trajectoires de rentrée en présence de perturbations aérologiques

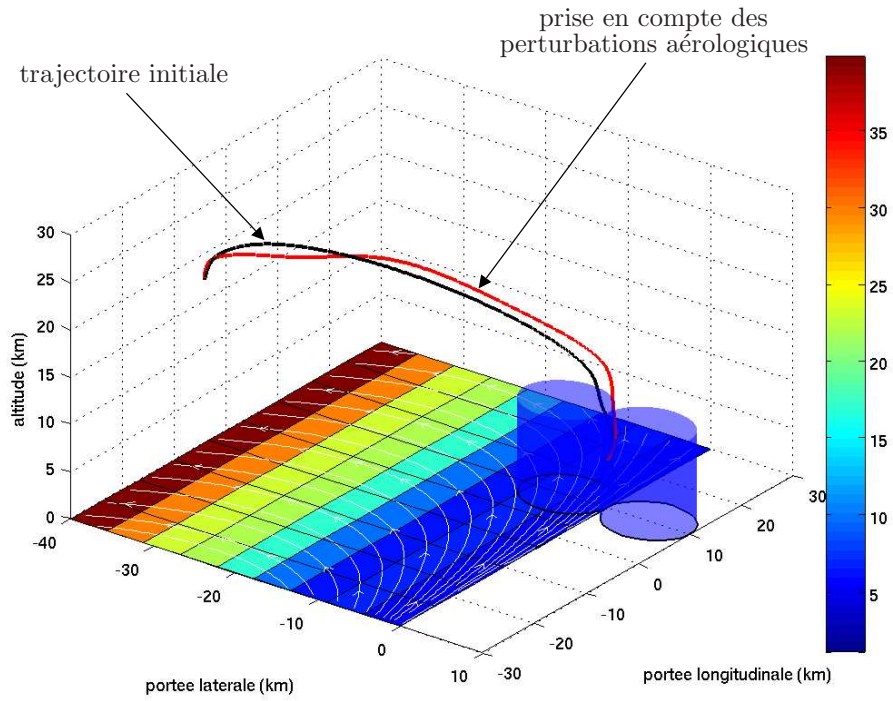


(a)

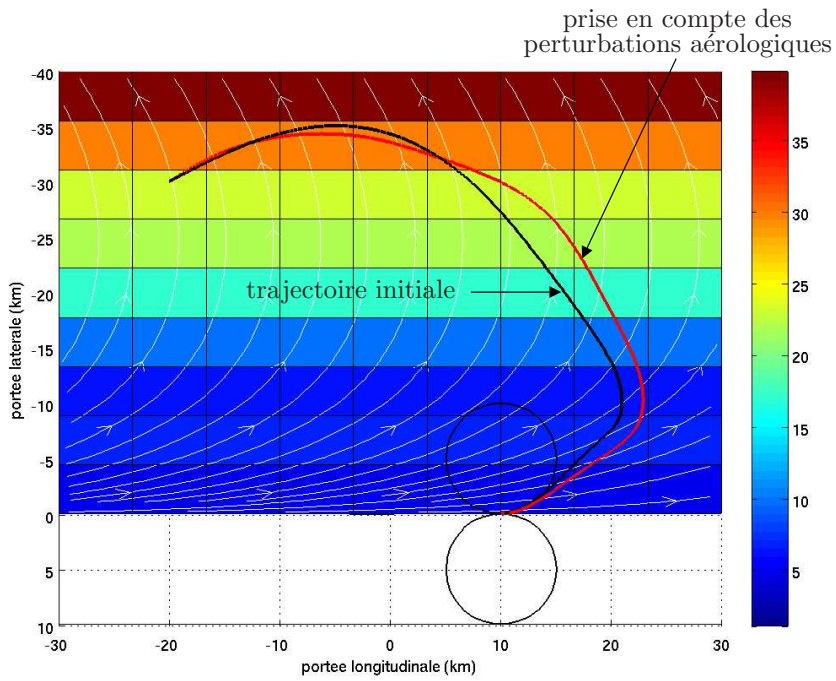


(b)

FIG. 3.29 – prise en compte de perturbations aérologiques dans le processus de planification de trajectoires.



(a)



(b)

FIG. 3.30 – prise en compte de perturbations aérodynamiques dans le processus de planification de trajectoires.

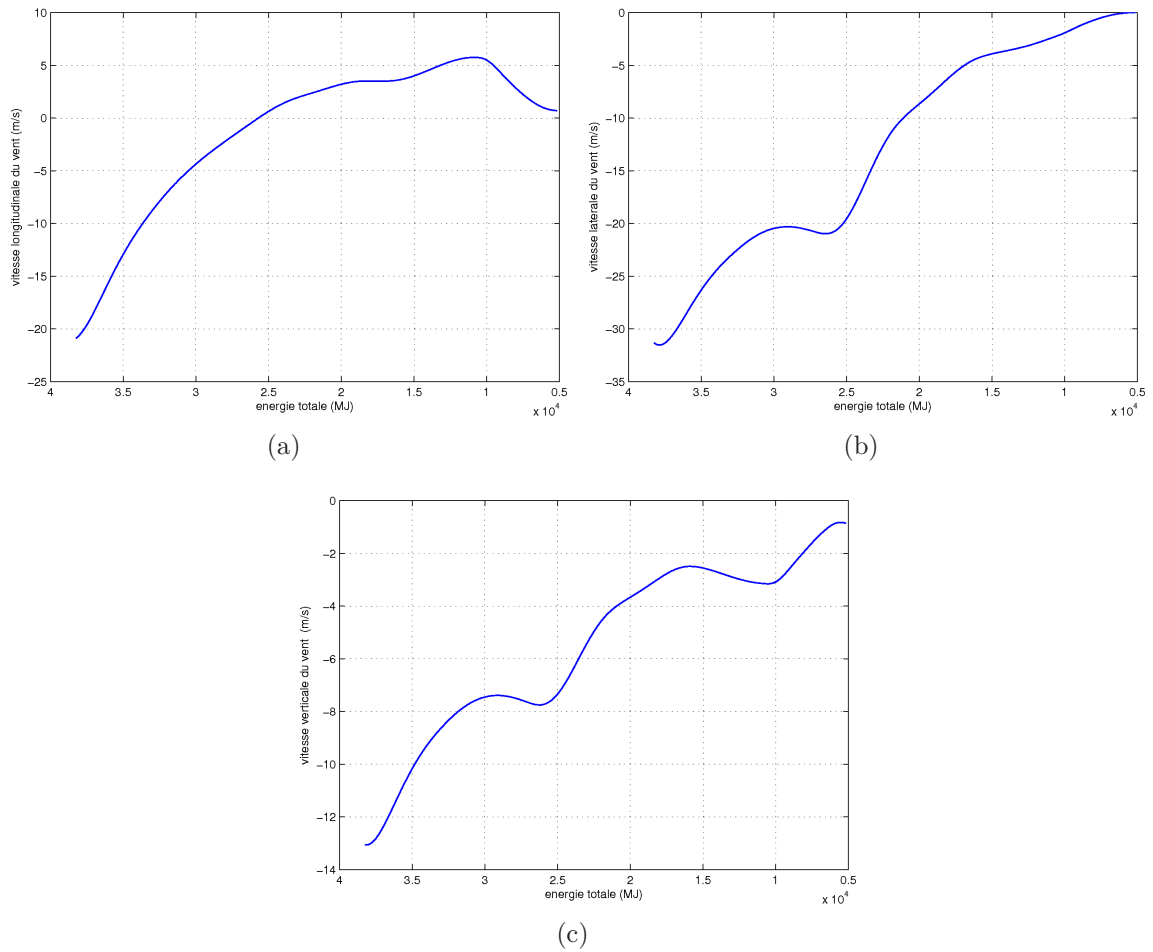


FIG. 3.31 – (a) vitesse de cisaillement de vent longitudinale (b) vitesse de cisaillement de vent latérale (c) vitesse de cisaillement de vent verticale.

A travers cette analyse, nous avons pu constater que la prise en compte de perturbations aérologiques dans le problème de commande optimale pouvait permettre de réduire l'énergie dépensée par le régulateur de poursuite de trajectoire et/ou la loi de pilotage afin de contrer les effets du vent. De plus, le temps CPU nécessaire pour résoudre le problème de commande optimale (3.90) étant encore très inférieur à la seconde<sup>62</sup>, cela montre clairement le potentiel de l'approche proposée pour tenir compte de champs de cisaillement de vent non uniformes et non stationnaires.

<sup>62</sup>du même ordre de grandeur que dans la section 3.3.2.

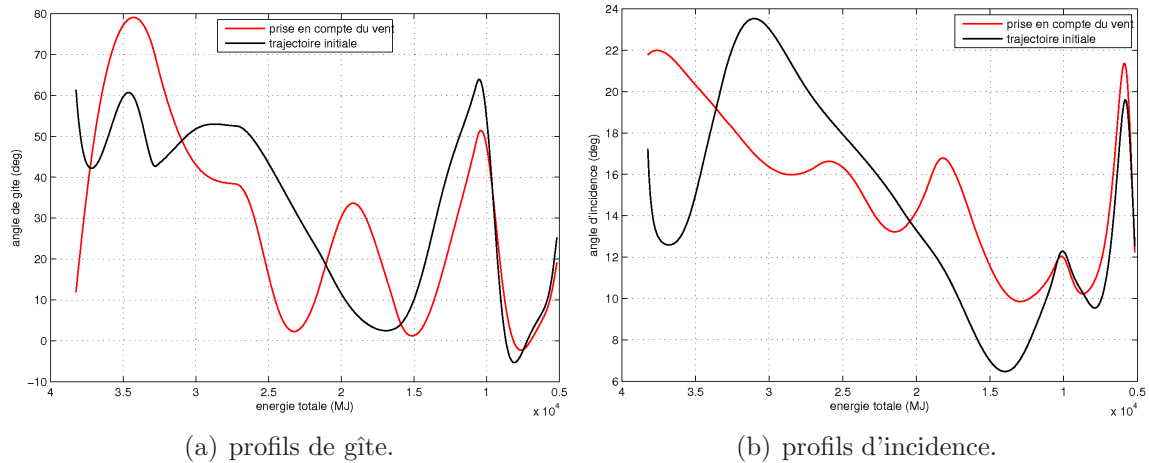


FIG. 3.32 – profils des consignes de guidage avec et sans perturbations aérologiques.

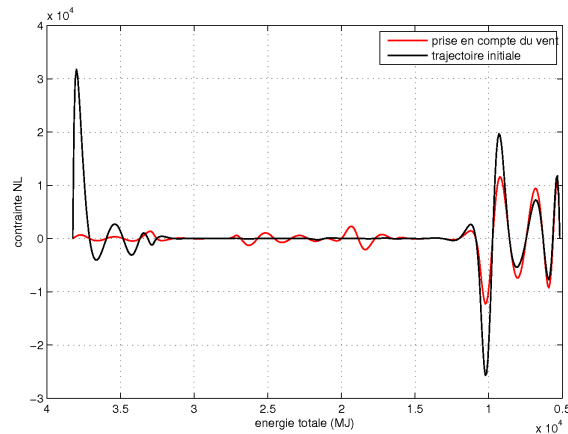


FIG. 3.33 – contrainte égalitaire de trajectoire avec et sans perturbations aérologiques ( $N$ ).

### 3.5 Convexification du problème de commande optimale

Dans cette section, nous allons appliquer la méthodologie d'annexion superellipsoïdale décrite dans la section 2.4 au problème de planification de trajectoires TAEM à bord. Rappelons d'abord très brièvement la démarche proposée. La première étape du processus consiste à approximer les différentes contraintes intervenant dans le problème de commande optimale par des formes géométriques convexes. Ensuite, les fonctions d'appartenance associées aux formes superquadriques sont intégrées dans le problème de commande optimale afin de le rendre convexe. Le nouveau problème de commande optimale est finalement discrétisé pour faire apparaître un problème de type NLP pouvant être résolu à bord du véhicule.

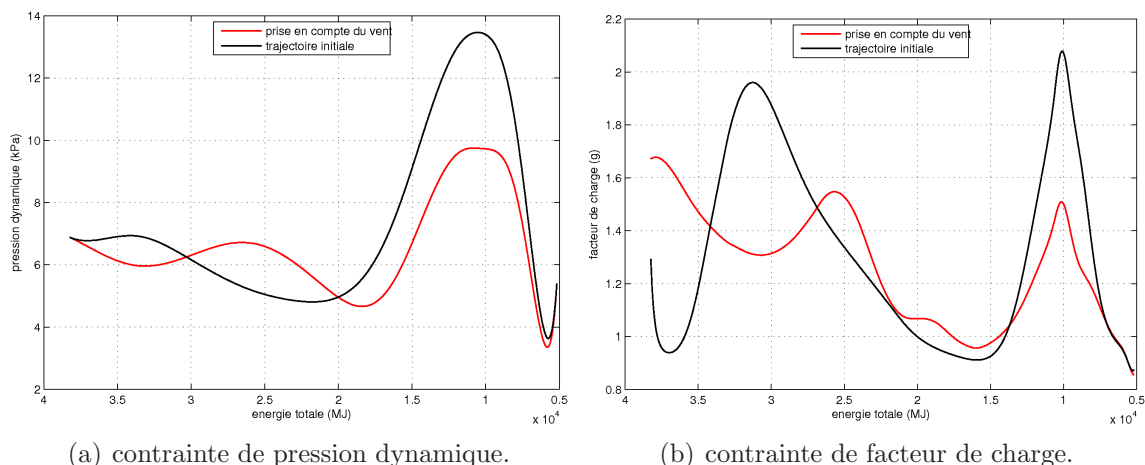


FIG. 3.34 – profils des contraintes mécaniques avec et sans perturbations aérologiques.

### 3.5.1 Approximation convexe intérieure des contraintes de trajectoire

Afin d'approximer les contraintes de trajectoire de manière convexe, nous considérons à nouveau le problème de commande optimale (3.44), paramétré en fonction de la durée totale libre de la trajectoire  $\lambda$ . Nous allons nous attacher à décrire le processus d'annexion superellipsoïdale associé à la contrainte de pression dynamique, les autres contraintes de trajectoires pouvant être approximées suivant le même schéma.

Considérons donc l'équation non linéaire associée à la contrainte de pression dynamique  $\bar{q}$ , exprimée en fonction des sorties plates, de leurs dérivées, ainsi que du paramètre de trajectoire  $\lambda$ . D'après le tableau 3.3, la pression dynamique ne doit pas excéder  $\bar{q}_{\max}=16$  kPa le long de la trajectoire TAEM. On a donc

$$0 \leq \frac{1}{2}\rho_0 \exp\left(-\frac{z_3}{H_0}\right) S \frac{\sqrt{z_1'^2 + z_2'^2 + z_3'^2}}{\lambda} \leq \bar{q}_{\max}. \quad (3.91)$$

D'un point de vue géométrique, il est possible de voir cette contrainte comme un volume généré par une famille de cercles centrés sur  $z_3$  se contractant de manière exponentielle (lorsque l'on ne tient pas compte de la contrainte de pente de trajectoire négative), la durée libre de trajectoire  $\lambda$  agissant alors comme un facteur d'homothétie de la forme sphérique. La figure 3.35 illustre ces considérations géométriques. Afin de réaliser une approximation intérieure convexe de la contrainte de pression dynamique, nous considérons un superellipsoïde d'ordre 5, ainsi que les transformations géométriques 5-D introduites dans la section 2.4. Notamment, d'après les considérations géométriques précédentes, la transformation de pincement linéaire sera effectuée le long de l'axe  $z_3$ , qui correspond à l'altitude du véhicule. L'ensemble des paramètres  $\Psi$  associés à cette forme superquadrique



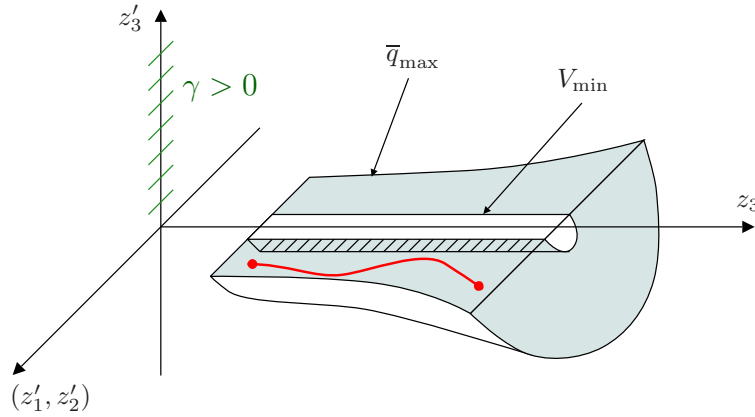


FIG. 3.35 – considérations géométriques sur la contrainte de pression dynamique.

s'écrit (voir équation (2.23))

$$\Psi = \left\{ \underbrace{a_1, \dots, a_5}_{\text{demi-grand axes}}, \underbrace{\varepsilon_1, \dots, \varepsilon_4}_{\text{par. de forme}}, \underbrace{\Phi_1, \dots, \Phi_{15}}_{\text{par. de rotation}}, \underbrace{d_1, \dots, d_5}_{\text{par. de translation}}, \underbrace{v_1, \dots, v_4}_{\text{par. de pincement}} \right\}. \quad (3.92)$$

De plus, du fait de la géométrie du superellipsoïde attendue (cf figure 3.35), les paramètres de rotation n'ont pas été considérés pour l'approximation de la contrainte de pression dynamique, i.e.,  $\Phi_i = 0$  pour  $i = 1, \dots, 15$ .

Maintenant que nous avons choisi une forme géométrique candidate, il est possible d'optimiser sa forme et son volume en utilisant un algorithme génétique (cf section 2.4). Le problème d'annexion superellipsoïdale est alors défini par 2.1. Les paramètres simples de réglage de l'algorithme génétique qui ont été retenus pour approximer la contrainte de pression dynamique sont indiqués ci-dessous :

- 100 individus répartis en 3 sous-populations,
- topologie de migration par réseau complet,
- nombre maximum de générations : 50,
- opérateur de sélection par tournoi avec domination (2.46) et classification linéaire des individus selon leur aptitude,
- opérateur de mutation paramétrique (cf algorithme 11), taux de mutation de 1% et indice de distribution  $\eta_c = 1$ ,
- opérateur de croisement binaire simulé SBX (cf algorithme 10), taux de croisement de 70% et indice de distribution  $\eta_m = 1$ ,
- Conflit de générations de 90% (nombre de nouveaux individus créés à chaque génération),
- Taux de réinsertion de 70% (nombre de descendants insérés dans la génération suivante),

L'évolution des aptitudes des individus au cours des générations est représentée sur la figure 3.36 avec une échelle logarithmique. L'évolution parallèle des trois sous-populations, due à la stratégie de migration utilisée, et l'échange de matériel génétique peut être claire-

ment observé jusqu'à atteindre l'optimum global. La forme géométrique convexe obtenue

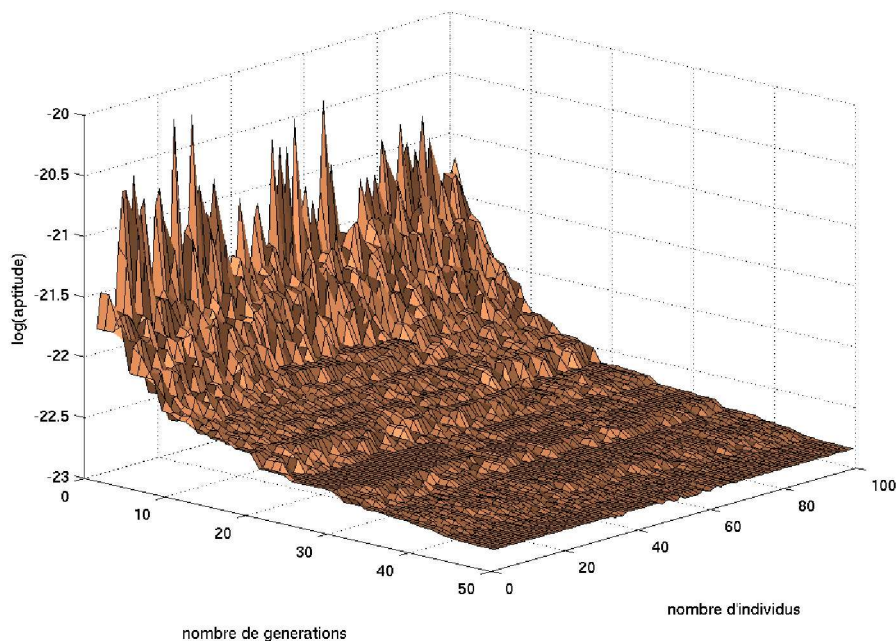


FIG. 3.36 – évolution de l'aptitude des individus au fil des générations.

est représentée sur la figure 3.37, en projection dans le repère  $(z'_1\lambda, z_3, z'_2\lambda)$ . De plus, les points associés à la paramétrisation angle-centre du superellipsoïde sont indiqués par des croix de couleur jaune. Le domaine admissible correspondant à la contrainte (3.91) est représenté au moyen d'un nuage de points dans le plan  $(z'_1\lambda, z_3, z'_2\lambda)$

La fonction d'appartenance  $\mathcal{F}_{\bar{q}}(\Psi^*, \bar{z})$ , définie dans la proposition 2.6, est alors donnée par

$$\begin{aligned} \mathcal{F}_{\bar{q}}(\Psi^*, \bar{z}) &= \left[ (0.8 \cdot 10^{-4} z_3 - 1.2)^{20} + \left( \frac{z'_1}{3.2 \cdot 10^4 + 5.3 z_3} \right)^{20} \right]^{0.1} + \left( \frac{z'_2}{3.5 \cdot 10^4 + 5.9 z_3} \right)^2 \\ &+ \left( \frac{z'_3}{3.1 \cdot 10^4 + 5.3 z_3} \right)^2 + \left( \frac{\lambda}{45.7 + 0.76 \cdot 10^{-2} z_3} \right)^2, \end{aligned} \quad (3.93)$$

où  $\Psi^*$  représente l'ensemble optimal des paramètres défini par (3.92), et  $\bar{z} = (z_3, z'_1, z'_2, z'_3, \lambda)$  est le vecteur des sorties plates et de leurs dérivées de dimension 5, exprimé dans l'espace des jets.

Le même processus de convexification a été appliqué à la contrainte d'angle de gîte, paramétrée en fonction des sorties plates, de leurs dérivées, et du paramètre libre de trajectoire  $\lambda$ , selon l'expression (3.38). Rappelons selon le tableau 3.4 que, durant la phase TAEM, l'angle de gîte doit rester borné dans l'intervalle  $[-80^\circ, 80^\circ]$ . Pour ce faire, nous considérons alors un superellipsoïde de dimension 7 équipé de toutes les transformations

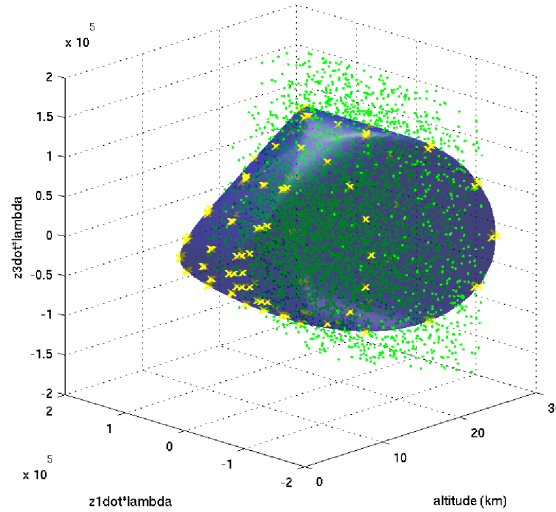


FIG. 3.37 – approximation de la contrainte de pression dynamique par une forme convexe. Le domaine admissible associé à la contrainte est représenté par un nuage de points de couleur verte. Les points associés à la paramétrisation angle-centre sont représentés par des croix de couleur jaune.

géométriques 7 –  $D$  définies dans la section 2.4, sauf du pincement linéaire qui ne présente pas d'intérêt dans ce cas précis. Les paramètres de réglage de l'algorithme génétique retenus sont approximativement les mêmes que pour la contrainte de pression dynamique. De plus, comme le nombre de variables  $\bar{z} = (z'_1, z'_2, z'_3, z''_1, z''_2, z''_3, \lambda)$  intervenant dans l'équation (3.38) engendre un nombre plus important de paramètres d'optimisation de la forme superquadrique, nous avons choisi 140 individus, répartis en 4 sous-populations, l'ensemble de la population évoluant sur 100 générations avant l'arrêt du processus d'optimisation.

Les formes superquadriques obtenues sont représentées sur la figure 3.38, en projection dans l'espace des vitesses  $(z'_1\lambda, z'_2\lambda, z'_3\lambda)$  et des accélérations  $(z''_1\lambda, z''_2\lambda, z''_3\lambda)$ , mis à l'échelle par le facteur  $\lambda$ .

Finalement, les autres contraintes de trajectoire telles que l'angle d'incidence (3.42), le facteur de charge (3.14) ou la contrainte associée à l'équation autonome (3.43) ont également été approximées par des formes superellipsoïdales en utilisant la même démarche.

**Remarque 3.24.** Dans le cadre du problème de planification de trajectoires TAEM, les contraintes de trajectoire s'expriment de manière très complexe et fortement non linéaire en fonction des sorties plates et de leurs dérivées. Aussi, nous ne sommes pas parvenu à trouver une unique forme géométrique convexe permettant d'approximer l'ensemble de ces contraintes. Nous utilisons alors le fait que l'intersection (non vide) d'enveloppes convexes est également une enveloppe convexe [159].

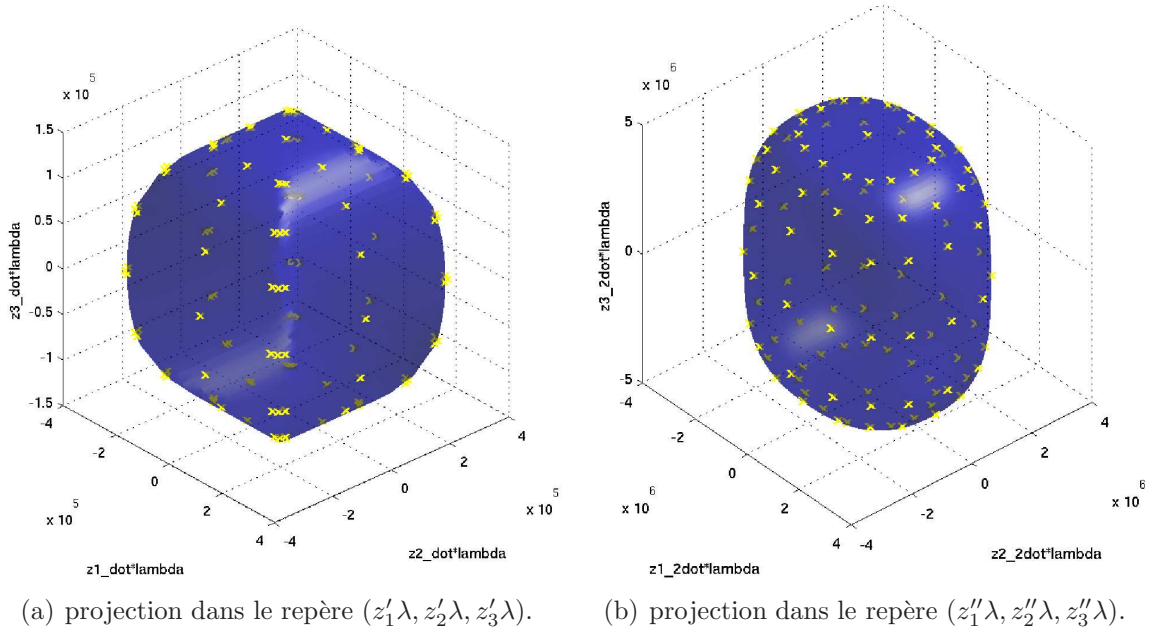


FIG. 3.38 – approximation du domaine de gîte admissible par des formes superellipsoidales.

### 3.5.2 Problème de commande optimale convexe

Une fois les différentes contraintes de trajectoires approximées par des formes convexes, il est possible de les intégrer dans le problème de commande optimale (3.44), où elles remplacent alors les contraintes initiales non convexes. Notons  $\mathcal{F}_{\bar{q}}$ ,  $\mathcal{F}_{\Gamma}$ ,  $\mathcal{F}_{\mu}$ ,  $\mathcal{F}_{\alpha}$  et  $\mathcal{F}_{\Lambda_{\tau}}$  les fonctions d'appartenance associées respectivement aux contraintes de pression dynamique (3.91), de facteur de charge (3.14), de gîte (3.38), d'incidence (3.42), et à l'observable autonome (3.43). De plus, nous ne considérons pas de fonctionnelle de coût particulière dans le cadre du problème de faisabilité considéré. Le problème de commande optimale (3.44) peut alors être réécrit sous la forme (2.47), tel que

$$\text{trouver } (\bar{z}(t), \lambda) \tag{3.94}$$

t.q.

$$\begin{aligned} \psi_x(\bar{z}(\tau_0), \lambda) &= x_0, \\ \psi_u(\bar{z}(\tau_0), \lambda) &= u_0, \\ 0 &\leq \mathcal{F}_{\Lambda_{\tau}}(\Psi^*, \bar{z}(t)) \leq 1, \quad \tau \in [\tau_0, \tau_f], \\ 0 &\leq \mathcal{F}_{\Gamma}(\Psi^*, \bar{z}(t)) \leq 1, \quad \tau \in [\tau_0, \tau_f], \\ 0 &\leq \mathcal{F}_{\bar{q}}(\Psi^*, \bar{z}(t)) \leq 1, \quad \tau \in [\tau_0, \tau_f], \\ 0 &\leq \mathcal{F}_{\mu}(\Psi^*, \bar{z}(t)) \leq 1, \quad \tau \in [\tau_0, \tau_f], \\ 0 &\leq \mathcal{F}_{\alpha}(\Psi^*, \bar{z}(t)) \leq 1, \quad \tau \in [\tau_0, \tau_f], \\ \psi_x(\bar{z}(\tau_f), \lambda) &= x_f, \\ \psi_u(\bar{z}(\tau_f), \lambda) &= u_f, \end{aligned}$$

où le vecteur  $\bar{z}$  est défini par  $\bar{z} = (z_1, z_2, z_3, \dot{z}_1, \dot{z}_2, \dot{z}_3, \ddot{z}_1, \ddot{z}_2, \ddot{z}_3)$  dans l'espace des jets d'ordre infini, et où le temps normalisé  $\tau$  vérifie  $\tau_0 = 0$  et  $\tau_f = 1$ .

La mission de rentrée considérée ici est la même que dans la section 3.3.2. En particulier, les conditions cinématiques initiales de trajectoire sont définies dans le tableau 3.3 et les conditions finales au point ALI dans le tableau 3.3. Les bornes choisies sur les consignes de guidage apparaissent dans le tableau 3.4.

Afin de résoudre le problème de commande optimale convexe (3.94), nous avons paramétré les variables d'optimisation  $z_1$ ,  $z_2$  et  $z_3$  par des B-splines d'ordre 7 et de multiplicité 3, comprenant chacune 5 intervalles, de manière à obtenir des dérivées temporelles d'ordre 1 et d'ordre 2 continues. Le paramètre  $\lambda$  a été modélisé, comme dans la section 3.3.2, par une B-spline comportant un seul intervalle, d'ordre 1 et de multiplicité nulle. De plus, les contraintes de trajectoire  $\mathcal{F}_{\bar{q}}$ ,  $\mathcal{F}_{\Gamma}$ ,  $\mathcal{F}_{\mu}$ ,  $\mathcal{F}_{\alpha}$  et  $\mathcal{F}_{\Lambda_\tau}$  sont vérifiées en 40 points de collocation régulièrement espacés dans l'intervalle  $[0, 1]$ , associé au temps normalisé  $\tau$ . Finalement, le nouvel OCP est discrétisé afin de le transformer en un problème de programmation non linéaire. Les variables de décision du problème NLP sont alors les paramètres de contrôle des B-splines.

Finalement, le problème NLP est résolu grâce au solveur NPSOL [44] en environ 0.2 s sur un ordinateur personnel équipé d'un processeur Intel Core 2 Duo cadencé à 2 Go. La trajectoire TAEM obtenue est représentée sur la figure 3.39. Le facteur de charge est également représenté le long de la trajectoire.

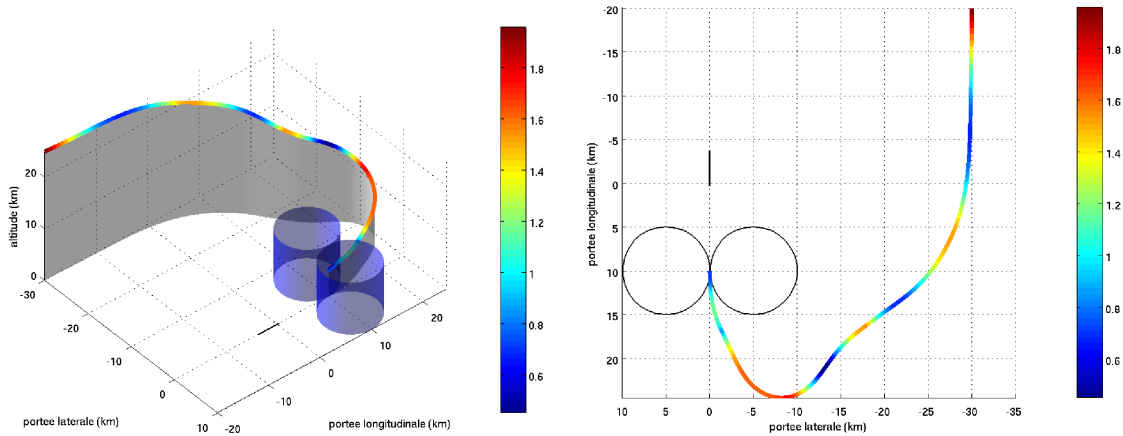


FIG. 3.39 – trajectoire TAEM obtenue par résolution du problème convexe.

Les profils des consignes de guidage  $\alpha$  et  $\mu$  correspondant à la trajectoire TAEM de référence sont représentés sur les figures 3.40(a) et 3.40(b).

De plus, la contrainte égalitaire de trajectoire (3.43) est représentée sur la figure 3.41.

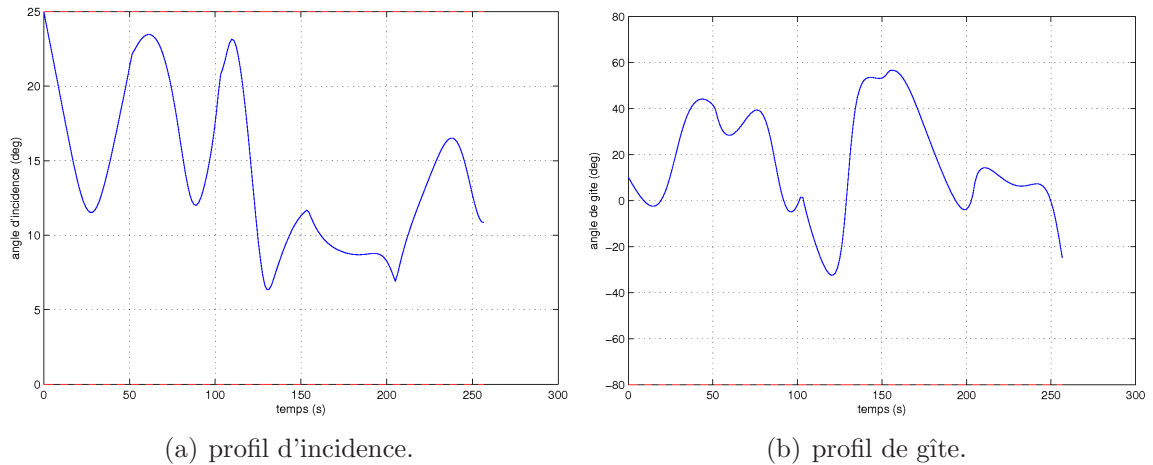


FIG. 3.40 – profils des consignes de guidage le long de la trajectoire TAEM de référence.

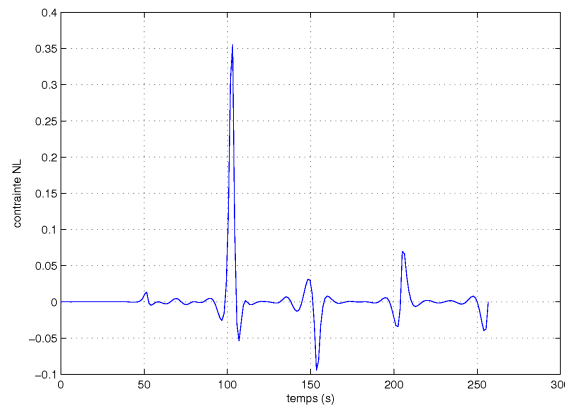


FIG. 3.41 – contrainte égalitaire de trajectoire ( $m/s^2$ ).

Les contraintes mécaniques subies par le véhicule durant la phase TAEM, c'est à dire la pression dynamique et le facteur de charge, sont représentées sur les figures 3.42(a) et 3.42(b). Clairement, la pression dynamique reste toujours inférieure à 16 kPa et le facteur de charge n'excède jamais 2.5 g.

En outre, il peut être intéressant de projeter la trajectoire TAEM planifiée dans les super-ellipsoïdes obtenus dans la section précédente. Ces derniers sont représentés sur la figure 3.43 pour la contrainte de pression dynamique, et sur la figure 3.44 dans le cas de la contrainte de gîte aérodynamique.

Comme la trajectoire TAEM projetée reste à chaque instant à l'intérieur des formes

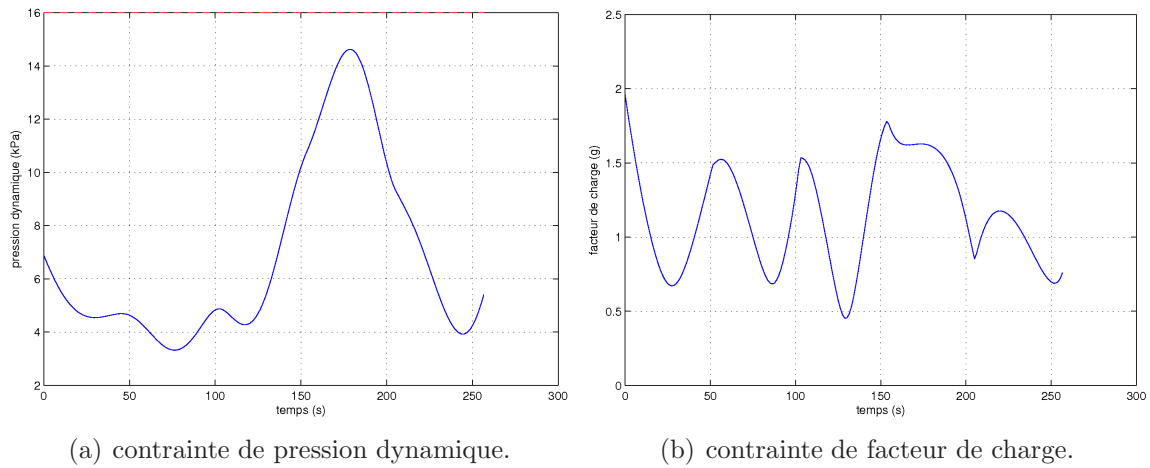


FIG. 3.42 – contraintes mécaniques le long de la trajectoire TAEM.

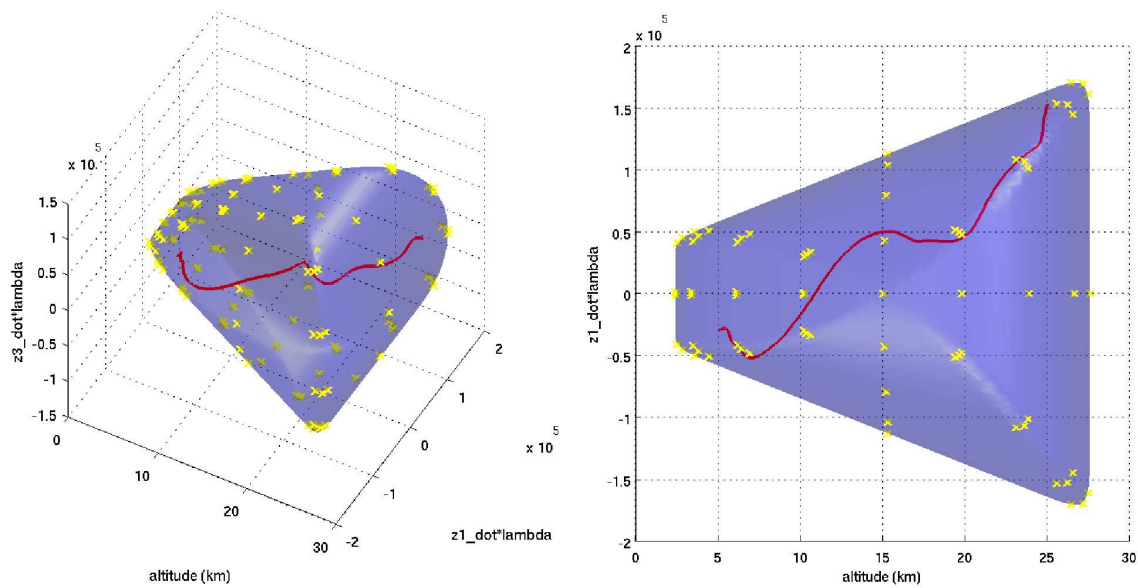


FIG. 3.43 – trajectoire TAEM projetée dans le superellipsoïde convexe de pression dynamique.

convexes représentées sur les figures 3.43 et 3.44, les fonctions d'appartenance associées aux contraintes de pression dynamique et d'angle de gîte sont respectées (voir l'OCP (3.94)). Le respect de ces fonctions d'appartenance garantit à son tour la satisfaction des contraintes considérées. Ces dernières sont représentées sur les figures 3.45(a) et 3.45(b). De manière évidente, les contraintes  $0 \leq \mathcal{F}_{\bar{q}}(\Psi^*, \bar{z}(t)) \leq 1$  et  $0 \leq \mathcal{F}_{\mu}(\Psi^*, \bar{z}(t)) \leq 1$  sont bien vérifiées.



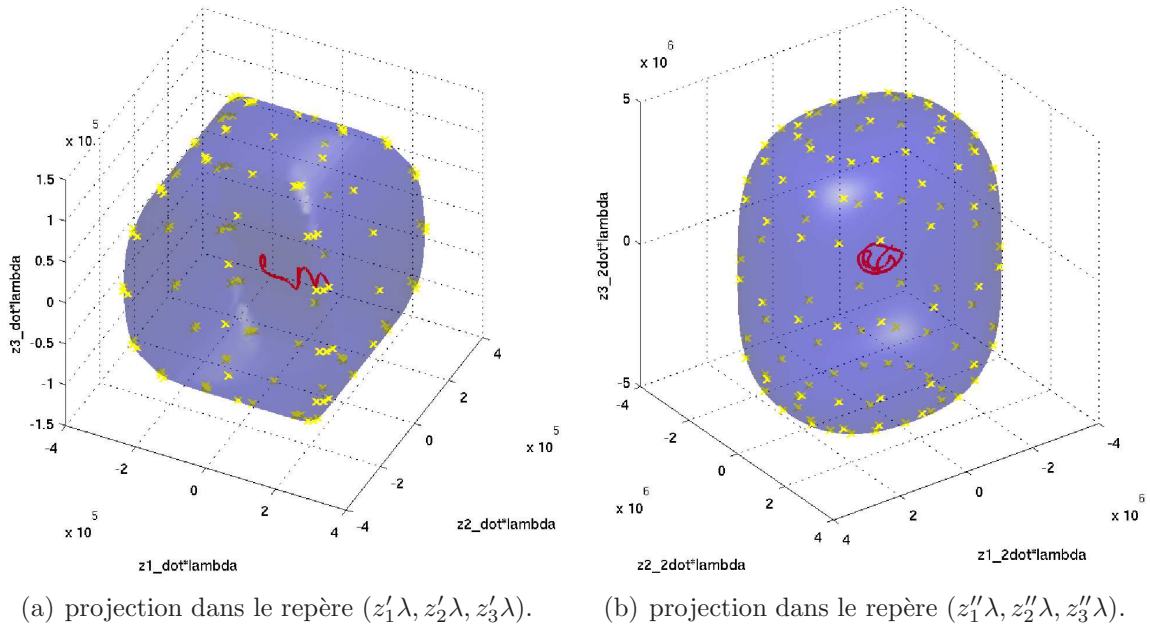


FIG. 3.44 – trajectoire TAEM projetée dans le superellipsoïde convexe de gîte aérodynamique.

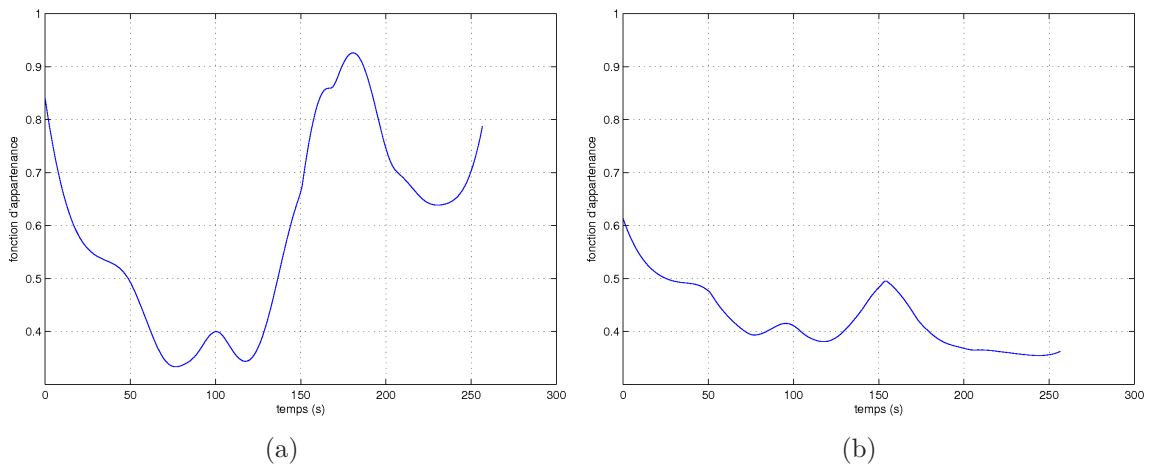


FIG. 3.45 – (a) fonction d'appartenance associée à la contrainte de pression dynamique (b) fonction d'appartenance associée à la contrainte de gîte.

En conclusion, la méthode proposée permet de transformer un problème de commande optimale décrit dans un cadre purement algébrique en un problème d'optimisation formulé dans un contexte géométrique. En effet, on cherche alors une trajectoire de rentrée atmosphérique inscrite dans une forme géométrique convexe, résultat de l'intersection d'un ensemble de formes superellipsoïdales. L'inscription de la trajectoire à l'intérieur de ces



formes géométriques garantit alors la satisfaction des contraintes de trajectoire associées au problème. Cependant, l'approximation convexe intérieure de ces contraintes induit un certain degré de conservatisme de la méthode qu'il serait intéressant de quantifier. Notamment, en présence de conditions cinématiques initiales dispersées, une trajectoire TAEM qualifiée d'admissible pour l'OCP initial peut alors devenir infaisable pour l'OCP convexe, si le processus d'annexion superellipsoïdale se révèle être trop conservateur. En particulier, cela peut se produire si la réduction du domaine admissible original, engendrée par le processus de convexification, se révèle être trop importante. Un autre axe d'amélioration de la méthode proposée concerne la méthode de résolution. En effet, l'OCP convexe jouit d'une propriété de convergence globale qui pourrait être mise à profit en résolvant le problème d'optimisation par des techniques de programmation semi-définies positives, basées sur la résolution d'inégalités matricielles linéaires (LMI).

## 3.6 Prise en compte de défaillances de gouvernes

Dans cette section, nous allons appliquer la méthodologie présentée dans la section 2.5.2, permettant de tenir compte de certaines défaillances de gouvernes pouvant potentiellement survenir durant la phase de vol TAEM. Seule la première solution envisagée sera décrite ici, c'est à dire celle fondée sur la prise en compte explicite des contraintes d'équilibrage statique dans le problème de commande optimale. La seconde méthode proposée dans la section 2.5.2, basée sur une modélisation hors ligne de l'enveloppe de vol admissible, est supposée fournir des résultats similaires, moyennant un temps de calcul plus faible pour déterminer une nouvelle trajectoire admissible à bord, le nombre de variables d'optimisation restant inchangé. En revanche, la première solution permet de traiter un plus grand nombre de situations défaillantes, sans nécessiter une analyse approfondie au sol de l'enveloppe de vol atteignable suite à une panne simple/multiple de gouvernes.

### 3.6.1 Expressions des coefficients de forces et de moments aérodynamiques de l'orbiteur STS-1

Le première étape consiste à modéliser de manière précise l'ensemble des coefficients de forces et de moments aérodynamiques du véhicule, en tenant compte de l'effet des gouvernes afin de pouvoir quantifier l'effet d'une défaillance sur la dynamique de l'appareil.

La base aérodynamique<sup>63</sup> de l'orbiteur STS-1 est extraite du manuel aérodynamique de référence [166], comprenant plus de 1600 pages. Nous allons commencer par négliger un ensemble de termes aérodynamiques non prédominants vis à vis de la trajectoire de vol considérée. Notamment, l'effet de sol, celui associé au déploiement des roues, les déformations aéroélastiques et les interactions avec le système de contrôle d'attitude RCS sont ignorés. Notons qu'en configuration lisse, les coefficients de forces et de moment sont calculés pour un braquage nominal de l'aérofrein de  $25^\circ$  ( $\delta_{sb} = 25^\circ$ ) et avec le body flap en position neutre ( $\delta_{bf} = 0^\circ$ ). De plus, les données relatives à la surface de référence, à

---

<sup>63</sup>disponible gratuitement sur le site de la NASA à l'adresse <http://ntrs.nasa.gov/>

la position du centre de gravité et aux moments et produits d'inertie de l'appareil sont indiqués dans le tableau 3.2. Sous ces hypothèses, les expressions des coefficients de forces et de moments aérodynamiques sont données ci-dessous.

**Coefficient total de portance  $C_{L_{tot}}$  :**

$$\begin{aligned} C_{L_{tot}}(\alpha, M, \delta) &= C_{L_0}(\alpha, M) + C_{L_{\delta_{bf}}}(\alpha, M, \delta_{bf}) + C_{L_{\delta_{sb}}}(\alpha, M, \delta_{sb}) \\ &+ \frac{1}{2} \left\{ \left[ K_{L_{\delta_e}}(\delta_{e_l}, M) C_{L_{\delta_e}}(\alpha, M, \delta_{e_{i_l}}) + (1 - K_{L_{\delta_e}}(\delta_{e_l}, M)) C_{L_{\delta_e}}(\alpha, M, \delta_{e_{o_l}}) \right] \right. \\ &+ \left. \left[ K_{L_{\delta_e}}(\delta_{e_r}, M) C_{L_{\delta_e}}(\alpha, M, \delta_{e_{i_r}}) + (1 - K_{L_{\delta_e}}(\delta_{e_r}, M)) C_{L_{\delta_e}}(\alpha, M, \delta_{e_{o_r}}) \right] \right\} \end{aligned} \quad (3.95)$$

**Coefficient total de traînée  $C_{D_{tot}}$  :**

$$\begin{aligned} C_{D_{tot}}(\alpha, M, \delta) &= C_{D_0}(\alpha, M) + \frac{1}{2} \left[ C_{D_{\delta_e}}(\alpha, M, \delta_{e_l}) + C_{D_{\delta_e}}(\alpha, M, \delta_{e_r}) \right] \\ &+ C_{D_{\delta_{bf}}}(\alpha, M, \delta_{bf}) + C_{D_{\delta_{sb}}}(\alpha, M, \delta_{sb}). \end{aligned} \quad (3.96)$$

**Coefficient total de force latérale  $C_{Y_{tot}}$  :**

$$\begin{aligned} C_{Y_{tot}}(\alpha, \beta, M, \delta) &= C_{Y_0}(\alpha, |\beta|, M) \text{sign}(\beta) + C_{Y_{\delta_{sb}}}(\alpha, M, \delta_{sb})\beta + C_{Y_{\delta_r}}(\alpha, M, \delta_{sb})\delta_r \\ &+ C_{Y_{\delta_e}}(\alpha, M, \delta_e)\beta + C_{Y_{\delta_a}}(\alpha, M, \delta_e) \left[ K_{Y_{\delta_e}}\delta_{a_i} + (1 - K_{Y_{\delta_e}})\delta_{a_o} \right]. \end{aligned} \quad (3.97)$$

**Coefficient total de moment de roulis  $C_{l_{tot}}$  :**

$$\begin{aligned} C_{l_{tot}}(\alpha, \beta, M, \delta) &= C_{l_0}(\alpha, |\beta|, M) \text{sign}(\beta) + C_{l_{\delta_{sb}}}(\alpha, M, \delta_{sb})\beta + C_{l_{\delta_r}}(\alpha, M, \delta_{sb})\delta_r \\ &+ C_{l_{\delta_a}}(\alpha, M, \delta_e) \left[ K_{l_{\delta_e}}(\delta_e, M)\delta_{a_i} + (1 - K_{l_{\delta_e}}(\delta_e, M))\delta_{a_o} \right] \\ &+ C_{l_{\delta_e}}(\alpha, M, \delta_e)\beta + C_{l_{\hat{r}}}(\alpha, M) \frac{rb}{2V} + C_{l_{\hat{p}}}(\alpha, M) \frac{pb}{2V}. \end{aligned} \quad (3.98)$$

**Coefficient total de moment de tangage  $C_{m_{tot}}$  :**

$$\begin{aligned} C_{m_{tot}}(\alpha, M, \delta) &= C_{m_0}(\alpha, M) + C_{m_{\delta_{bf}}}(\alpha, M, \delta_{bf}) + C_{m_{\delta_{sb}}}(\alpha, M, \delta_{sb}) \\ &+ \frac{1}{2} \left\{ \left[ K_{m_{\delta_e}}(\delta_{e_l}, M) C_{m_{\delta_e}}(\alpha, M, \delta_{e_{i_l}}) + (1 - K_{m_{\delta_e}}(\delta_{e_l}, M)) C_{m_{\delta_e}}(\alpha, M, \delta_{e_{o_l}}) \right] \right. \\ &+ \left. \left[ K_{m_{\delta_e}}(\delta_{e_r}, M) C_{m_{\delta_e}}(\alpha, M, \delta_{e_{i_r}}) + (1 - K_{m_{\delta_e}}(\delta_{e_r}, M)) C_{m_{\delta_e}}(\alpha, M, \delta_{e_{o_r}}) \right] \right\} \\ &+ \left[ C_{m_{\hat{q}}}(\alpha, M) \left( K_{m_{\delta_{sb}}} \frac{\delta_{sb} - 25}{75} + (1 - K_{m_{\delta_{sb}}}) \frac{\delta_{bf}}{13.75} \right) \Delta C_{m_{\hat{q}}}(\alpha, M) \right] \frac{q\bar{c}}{2V}. \end{aligned} \quad (3.99)$$

**Coefficient total de moment de lacet  $C_{n_{tot}}$  :**

$$\begin{aligned} C_{n_{tot}}(\alpha, \beta, M, \delta) &= C_{n_0}(\alpha, |\beta|, M) \text{sign}(\beta) + C_{n_{\delta_{sb}}}(\alpha, M, \delta_{sb})\beta + C_{n_{\delta_r}}(\alpha, M, \delta_{sb})\delta_r \\ &+ C_{n_{\delta_a}}(\alpha, M, \delta_e) \left[ K_{n_{\delta_e}}(\delta_e, M)\delta_{a_i} + (1 - K_{n_{\delta_e}}(\delta_e, M))\delta_{a_o} \right] \\ &+ C_{n_{\delta_e}}(\alpha, M, \delta_e)\beta + C_{n_{\hat{r}}}(\alpha, M) \frac{rb}{2V} + C_{n_{\hat{p}}}(\alpha, M) \frac{pb}{2V}. \end{aligned} \quad (3.100)$$

où  $\delta_{sb}$  désigne de braquage de l'aérofrein,  $\delta_r$  celui de la gouverne de direction et  $\delta_{bf}$  celui du body flap. De plus, les braquages des élévons sont calculés de la manière suivante :

$$\delta_e = \frac{1}{2}(\delta_{e_i} + \delta_{e_o}) = \frac{1}{2}(\delta_{e_{i_l}} + \delta_{e_{i_r}}) \quad (3.101)$$

$$\delta_{e_i} = \frac{1}{2}(\delta_{e_{i_l}} + \delta_{e_{i_r}}) \quad (3.102)$$

$$\delta_{e_o} = \frac{1}{2}(\delta_{e_{o_l}} + \delta_{e_{o_r}}) \quad (3.103)$$

$$\delta_{e_l} = \frac{1}{2}(\delta_{e_{i_l}} + \delta_{e_{o_l}}) = \delta_e + \delta_a \quad (3.104)$$

$$\delta_{e_r} = \frac{1}{2}(\delta_{e_{i_r}} + \delta_{e_{o_r}}) = \delta_e - \delta_a \quad (3.105)$$

$$(3.106)$$

où  $\delta_{e_{i_l}}$  représente le braquage de l'élévon intérieur gauche,  $\delta_{e_{i_r}}$  celui de l'élévon intérieur droit,  $\delta_{e_{o_l}}$  le braquage de l'élévon extérieur gauche et  $\delta_{e_{o_r}}$  celui de l'élévon extérieur droit. Le terme  $\delta_e$  représente la contribution des gouvernes sur la profondeur et  $\delta_a$  désigne l'effet des gouvernes sur le gauchissement de l'appareil, tel que

$$\delta_a = \frac{1}{2}(\delta_{e_l} - \delta_{e_r}) = \frac{1}{2}(\delta_{a_i} + \delta_{a_o}) \quad (3.107)$$

$$\delta_{a_i} = \frac{1}{2}(\delta_{e_{i_l}} - \delta_{e_{i_r}}) \quad (3.108)$$

$$\delta_{a_o} = \frac{1}{2}(\delta_{e_{o_l}} - \delta_{e_{o_r}}) \quad (3.109)$$

$$(3.110)$$

De plus, le signe de l'angle de dérapage  $\beta$  est calculé avec la convention suivante :

$$\text{sign}(\beta) = \begin{cases} +1 & \text{si } \beta \geq 0 \\ -1 & \text{si } \beta < 0 \end{cases} \quad (3.111)$$

Nous résumons ci-après les différents termes aérodynamiques apparaissant dans les expressions (3.95)-(3.100).

Les termes aérodynamiques  $C_{L_0}(\alpha, M)$ ,  $C_{D_0}(\alpha, M)$ ,  $C_{Y_0}(\alpha, |\beta|, M)$ ,  $C_{l_0}(\alpha, |\beta|, M)$ ,  $C_{m_0}(\alpha, M)$  et  $C_{n_0}(\alpha, |\beta|, M)$  représentent les coefficients de forces et de moments en configuration lisse, pour  $\delta_{sb} = 25^\circ$  et  $\delta_{sb} = 0^\circ$ .

L'effet des élévons sur l'aérodynamique du véhicule est représenté au moyen des termes  $C_{L_{\delta_e}}(\alpha, M, \delta_e)$ ,  $C_{D_{\delta_e}}(\alpha, M, \delta_e)$ ,  $C_{Y_{\delta_e}}(\alpha, M, \delta_e)$ ,  $C_{l_{\delta_e}}(\alpha, M, \delta_e)$ ,  $C_{m_{\delta_e}}(\alpha, M, \delta_e)$  et  $C_{n_{\delta_e}}(\alpha, M, \delta_e)$ , celui des ailerons par  $C_{Y_{\delta_a}}(\alpha, M, \delta_e)$ ,  $C_{l_{\delta_a}}(\alpha, M, \delta_e)$  et  $C_{n_{\delta_a}}(\alpha, M, \delta_e)$ .

L'effet de l'aérofrein est représenté par les coefficients  $C_{L_{\delta_{sb}}}(\alpha, M, \delta_{sb})$ ,  $C_{D_{\delta_{sb}}}(\alpha, M, \delta_{sb})$ ,  $C_{Y_{\delta_{sb}}}(\alpha, M, \delta_{sb})$ ,  $C_{l_{\delta_{sb}}}(\alpha, M, \delta_{sb})$ ,  $C_{m_{\delta_{sb}}}(\alpha, M, \delta_{sb})$  et  $C_{n_{\delta_{sb}}}(\alpha, M, \delta_{sb})$ , celui de la gouverne de direction par  $C_{Y_{\delta_r}}(\alpha, M, \delta_{sb})$ ,  $C_{l_{\delta_r}}(\alpha, M, \delta_{sb})$  et  $C_{n_{\delta_r}}(\alpha, M, \delta_{sb})$ .

L'influence du body flap sur l'aérodynamique du véhicule est modélisée par le biais des termes  $C_{L_{\delta_{bf}}}(\alpha, M, \delta_{bf})$ ,  $C_{D_{\delta_{bf}}}(\alpha, M, \delta_{bf})$ ,  $C_{Y_{\delta_{bf}}}(\alpha, M, \delta_{bf})$ ,  $C_{l_{\delta_{bf}}}(\alpha, M, \delta_{bf})$ ,  $C_{m_{\delta_{bf}}}(\alpha, M, \delta_{bf})$  et  $C_{n_{\delta_{bf}}}(\alpha, M, \delta_{bf})$ .

L'effet des vitesses de roulis, tangage et lacet  $(p, q, r)$  est représenté par les coefficients  $C_{m_{\hat{q}}}(\alpha, M)$ ,  $\Delta C_{m_{\hat{q}}}$ ,  $C_{l_{\hat{r}}}(\alpha, M)$ ,  $C_{l_{\hat{p}}}(\alpha, M)$ ,  $C_{n_{\hat{r}}}(\alpha, M)$  et  $C_{n_{\hat{p}}}(\alpha, M)$ , où  $\hat{p} = \frac{pb}{2V}$ ,  $\hat{q} = \frac{q\bar{c}}{2V}$  et  $\hat{r} = \frac{rb}{2V}$ .

Finalement, presque tous les coefficients aérodynamiques (sauf  $C_{D_{tot}}$ ) font intervenir des coefficients d'efficacité de gouvernes  $K_{L_{\delta_e}}(\delta_e, M)$ ,  $K_{l_{\delta_e}}(\delta_e, M)$ ,  $K_{m_{\delta_e}}(\delta_e, M)$  et  $K_{n_{\delta_e}}(\delta_e, M)$ . De plus, les coefficients d'efficacité  $K_{Y_{\delta_e}}$  et  $K_{m_{\delta_{sb}}}$  sont constants et valent respectivement  $K_{Y_{\delta_e}} = 0.6$  et  $K_{m_{\delta_{sb}}} = 0.5$ .

Comme nous cherchons à modéliser cette base aérodynamique sous forme analytique, nous allons d'abord simplifier les expressions (3.95)-(3.100) en fonction de certaines considérations. Premièrement, en accord avec la méthodologie présentée dans la section 2.5.1 pour calculer des cartes de trimmabilité statique, on se ramène au cas d'un vol symétrique, et on pose alors  $\beta = 0$  et  $\delta_r = 0$ . Ensuite, afin de simplifier les expressions précédentes des coefficients aérodynamiques, on choisit des coefficients d'efficacité de gouvernes moyens égaux à 0.5, i.e.,  $K_{L_{\delta_e}} = K_{Y_{\delta_e}} = K_{l_{\delta_e}} = K_{m_{\delta_e}} = K_{n_{\delta_e}} = 0.5$ . Finalement, comme on ne cherche pas à synthétiser une loi de pilotage, on peut prendre des vitesses de roulis, tangage et lacet nulles, i.e.,  $(p, q, r) = (0, 0, 0)$ , ce qui annule les termes  $C_{m_{\hat{q}}}(\alpha, M)$ ,  $\Delta C_{m_{\hat{q}}}$ ,  $C_{l_{\hat{r}}}(\alpha, M)$ ,  $C_{l_{\hat{p}}}(\alpha, M)$ ,  $C_{n_{\hat{r}}}(\alpha, M)$  et  $C_{n_{\hat{p}}}(\alpha, M)$ . On obtient alors les expressions simplifiées suivantes :

**Coefficient total de portance  $C_{L_{tot}}$  :**

$$C_{L_{tot}}(\alpha, M, \delta) = C_{L_0}(\alpha, M) + C_{L_{\delta_e}}(\alpha, M, \delta_e) + C_{L_{\delta_{bf}}}(\alpha, M, \delta_{bf}) + C_{L_{\delta_{sb}}}(\alpha, M, \delta_{sb}). \quad (3.112)$$

**Coefficient total de traînée  $C_{D_{tot}}$  :**

$$C_{D_{tot}}(\alpha, M, \delta) = C_{D_0}(\alpha, M) + C_{D_{\delta_e}}(\alpha, M, \delta_e) + C_{D_{\delta_{sb}}}(\alpha, M, \delta_{sb}). \quad (3.113)$$

**Coefficient total de force latérale  $C_{Y_{tot}}$  :**

$$C_{Y_{tot}}(\alpha, \beta, M, \delta) = C_{Y_{\delta_a}}(\alpha, M, \delta_e)\delta_a. \quad (3.114)$$

**Coefficient total de moment de roulis  $C_{l_{tot}}$  :**

$$C_{l_{tot}}(\alpha, \beta, M, \delta) = C_{l_{\delta_a}}(\alpha, M, \delta_e)\delta_a. \quad (3.115)$$

**Coefficient total de moment de tangage  $C_{m_{tot}}$  :**

$$C_{m_{tot}}(\alpha, M, \delta) = C_{m_0}(\alpha, M) + C_{m_{\delta_e}}(\alpha, M, \delta_e) + C_{m_{\delta_{bf}}}(\alpha, M, \delta_{bf}) + C_{m_{\delta_{sb}}}(\alpha, M, \delta_{sb}). \quad (3.116)$$

Coefficient total de moment de lacet  $C_{n_{tot}}$  :

$$C_{n_{tot}}(\alpha, \beta, M, \delta) = C_{n_{\delta_a}}(\alpha, M, \delta_e)\delta_a. \quad (3.117)$$

Après analyse des coefficients aérodynamiques  $C_{Y_{\delta_a}}$ ,  $C_{l_{\delta_a}}$  et  $C_{n_{\delta_a}}$ , représentés respectivement sur les figures 3.46(a), 3.46(b) et 3.46(c), la satisfaction simultanée des équations simplifiées (3.114), (3.115) et (3.117) impose que  $\delta_a = 0$ . La contrainte d'équilibrage statique du véhicule en mode latéral se résumera donc à vérifier que  $\delta_a = 0$ .

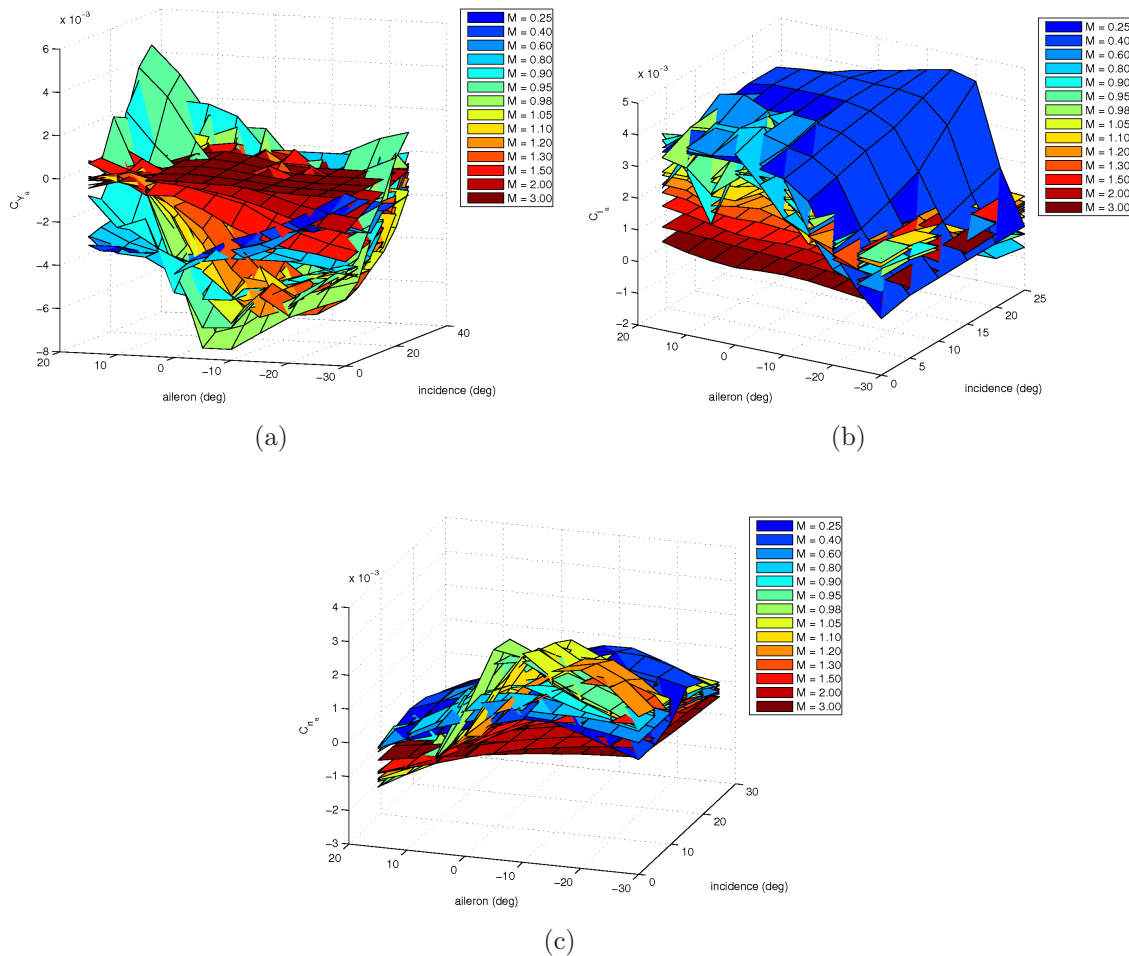


FIG. 3.46 – (a) effet des ailerons sur le coefficient de force latérale (b) effet des ailerons sur le coefficient de moment de roulis (c) effet des ailerons sur le coefficient de moment de lacet.

Finalement, le coefficient de portance peut être simplifié afin de faciliter l'inversion dynamique du modèle<sup>64</sup> dans la suite. Notamment, comparé à l'influence du nombre de Mach

<sup>64</sup>plus précisément, l'obtention de l'expression analytique de l'angle d'incidence  $\alpha$  s'en trouvera facilitée dans la section 3.6.3.

et du braquage des gouvernes, l'effet de l'incidence peut être négligé dans les termes de portance  $C_{L\delta_e}$ ,  $C_{L\delta_{bf}}$  et  $C_{L\delta_{sb}}$  (représentés sur les figures 3.47(a), 3.47(b) et 3.47(c)).

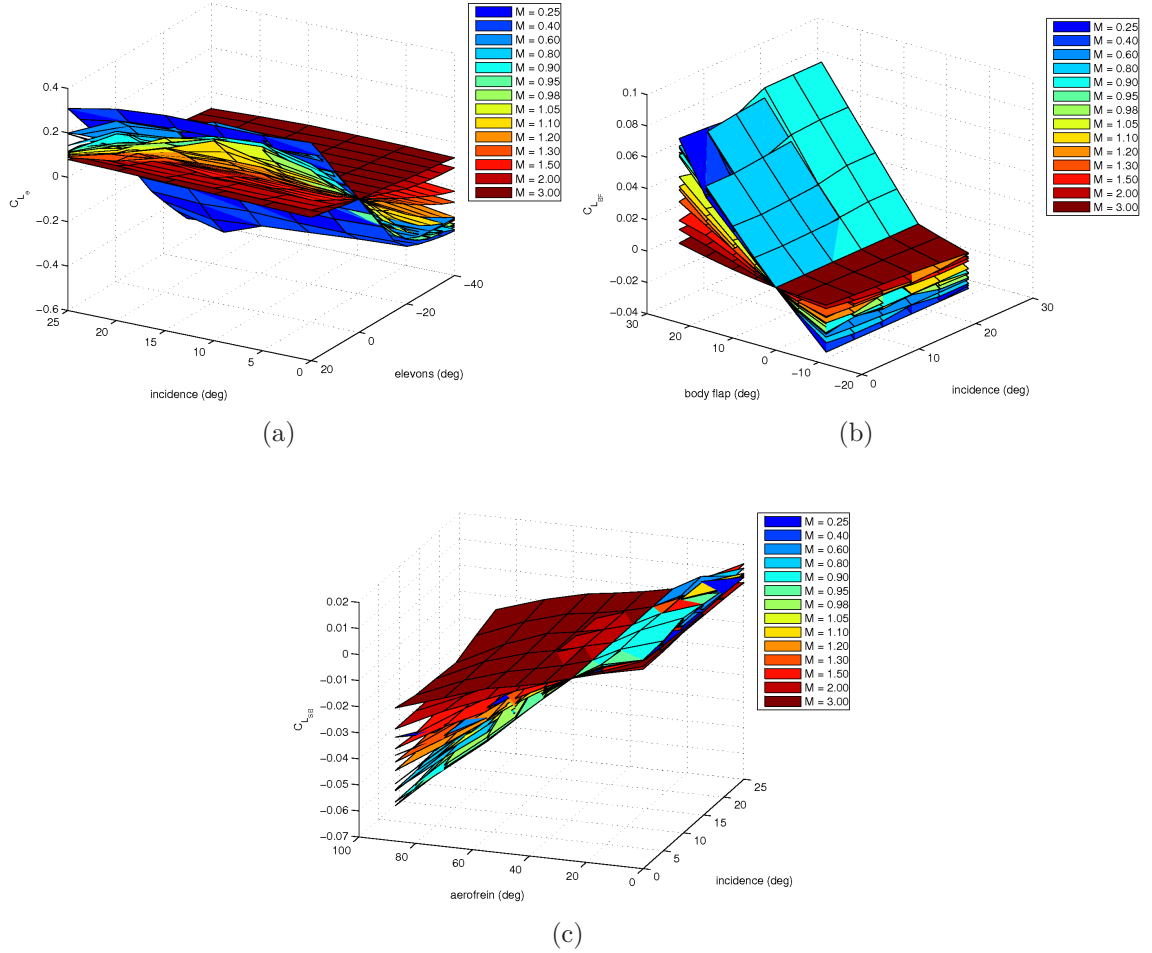


FIG. 3.47 – (a) effet des élévons sur le coefficient de portance (b) effet du body flap sur le coefficient de portance (c) effet de l'aérofrein sur le coefficient de portance.

On obtient alors les expressions finales des coefficients de forces aérodynamiques que nous avons retenu dans le cadre de cette étude. Les expressions des coefficients de forces sont alors données par

$$C_{L_{tot}}(\alpha, M, \delta) = C_{L_0}(\alpha, M) + C_{L\delta_e}(M, \delta_e) + C_{L\delta_{bf}}(M, \delta_{bf}) + C_{L\delta_{sb}}(M, \delta_{sb}) \quad (3.118)$$

$$C_{D_{tot}}(\alpha, M, \delta) = C_{D_0}(\alpha, M) + C_{D\delta_e}(\alpha, M, \delta_e) + C_{D\delta_{sb}}(\alpha, M, \delta_{sb}), \quad (3.119)$$

$$C_{Y_{tot}}(\alpha, \beta, M, \delta) = 0. \quad (3.120)$$

### 3.6.2 Génération des cartes de trimmabilité statique

La génération de cartes d'équilibrage statique n'est pas obligatoire dans le cadre de la première méthode de replanification de trajectoire tolérante aux défauts que nous avons proposée dans la section 2.5.2, les contraintes d'équilibrage étant directement incluses dans le problème de commande optimale. Cependant, la connaissance de ces cartes nous permettra de valider notre approche, et de vérifier *a posteriori* que la trajectoire TAEM planifiée se trouve bien dans le corridor de vol Mach- $\alpha$  admissible.

En utilisant les expressions des coefficients de moment définis dans la section précédente, le problème d'équilibrage statique 2.2 défini dans la section 2.5.1 se réécrit, en chaque point  $(\alpha_i, M_j)$  du domaine de définition de la base aérodynamique

$$\min_{\delta} J_D = \min_{\delta} \left\| \begin{bmatrix} T_{(l,n)\delta_{i,j}}(\delta_{i,j}) \\ C_{m\delta_{i,j}}(\alpha_i, M_j, \delta_{i,j}) \end{bmatrix} - \begin{bmatrix} 0 \\ -C_{m_{0i,j}}(\alpha_i, M_j) \end{bmatrix} \right\|_1 \quad (3.121)$$

t.q.

$$\underline{\delta} \leq \delta \leq \bar{\delta},$$

où  $T_{(l,n)\delta_{i,j}}$  représente la contrainte d'équilibrage latérale du véhicule, telle que

$$T_{(l,n)\delta_{i,j}}(\delta_{i,j}) = \delta_a = \frac{1}{4}(\delta_{e_{i_l}} - \delta_{e_{i_r}} + \delta_{e_{o_l}} - \delta_{e_{o_r}}), \quad (3.122)$$

et

$$C_{m\delta_{i,j}}(\alpha_i, M_j, \delta_{i,j}) = C_{m\delta_e}(\alpha_i, M_j, \delta_e) + C_{m\delta_{bf}}(\alpha_i, M_j, \delta_{bf}) + C_{m\delta_{sb}}(\alpha_i, M_j, \delta_{sb}).$$

De plus, pour la navette STS-1, le vecteur  $\delta$  est défini par

$$\delta = (\delta_{e_{i_l}}, \delta_{e_{i_r}}, \delta_{e_{o_l}}, \delta_{e_{o_r}}, \delta_{bf}, \delta_{sb})^T$$

et les limites de braquages sont définies par

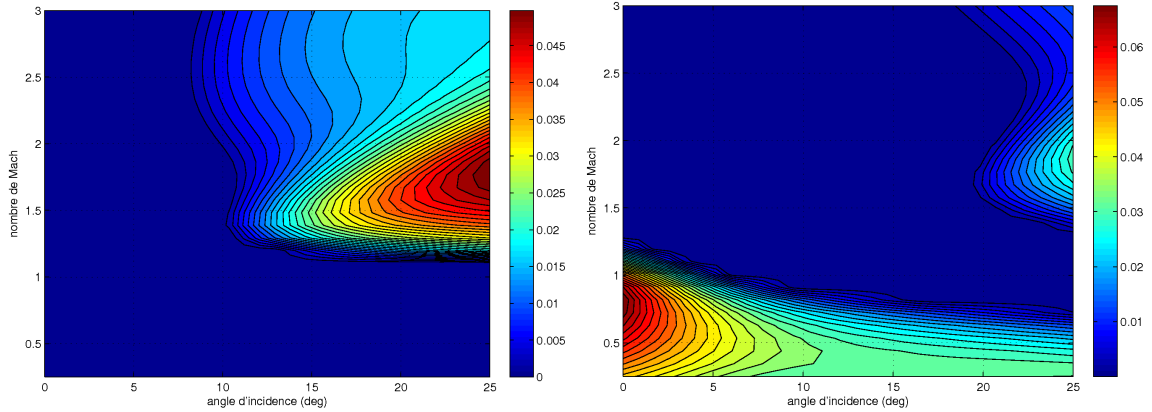
$$\begin{aligned} \bar{\delta} &= (\bar{\delta}_{e_{i_l}}, \bar{\delta}_{e_{i_r}}, \bar{\delta}_{e_{o_l}}, \bar{\delta}_{e_{o_r}}, \bar{\delta}_{bf}, \bar{\delta}_{sb})^T, \\ \underline{\delta} &= (\underline{\delta}_{e_{i_l}}, \underline{\delta}_{e_{i_r}}, \underline{\delta}_{e_{o_l}}, \underline{\delta}_{e_{o_r}}, \underline{\delta}_{bf}, \underline{\delta}_{sb})^T, \end{aligned}$$

où les limites de braquage minimales et maximales de chaque surface de contrôle aérodynamique sont indiquées dans le tableau 3.1.

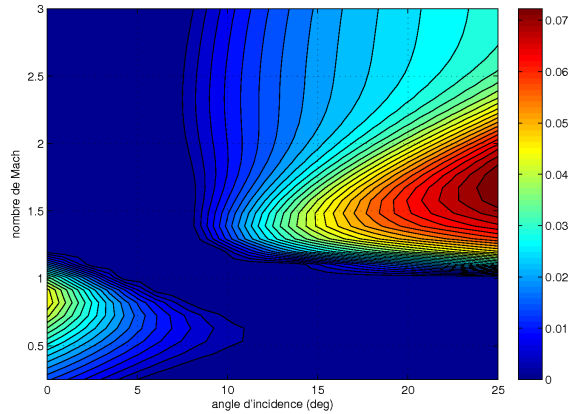
Tous les coefficients de forces et de moments intervenant dans le problème d'équilibrage (3.121) ont été modélisés grâce à des réseaux de neurones statiques, en suivant la démarche décrite dans la section 3.3.1. Le problème de programmation non linéaire (3.121) est résolu en utilisant le solveur NPSOL [44] sous Matlab, pour une grille de valeurs d'incidence  $\alpha \in [0^\circ, 25^\circ]$  et de nombres de Mach  $M \in [0.25, 3]$ . En outre, l'estimé initial du problème NLP est choisi tel que

$$\delta_0 = (0, 0, 0, 0, 0, 25)^T.$$

A chaque point  $(\alpha_i, M_j)$  de la grille Mach- $\alpha$  correspond un temps CPU moyen de 11 ms sur un ordinateur personnel équipé d'un processeur Intel Core 2 Duo cadencé à 2 GHz. Nous donnons ci-dessous les cartes d'équilibrage associées à quelques configurations de gouvernes défaillantes. Le bandeau coloré sur la droite représente les valeurs de déficience d'équilibrage  $\text{def}(i, j)$ , comme indiqué dans l'algorithme 12.



(a) carte d'équilibrage statique avec l'élevon ex- (b) carte d'équilibrage statique avec l'élevon exté-  
térieur gauche bloqué à  $17^\circ$  et l'aérofrein bloqué rieur gauche bloqué à  $-35^\circ$  et le body flap bloqué  
à  $0^\circ$ . à  $22.5^\circ$ .



(c) carte d'équilibrage statique avec l'élevon inté-  
rieur gauche bloqué à  $10^\circ$ , l'élevon intérieur droit  
bloqué à  $-20^\circ$ , le body flap bloqué à  $10^\circ$  et l'aé-  
rofrein bloqué en position neutre  $\delta_{sb} = 25^\circ$ .

FIG. 3.48 – exemples de cartes d'équilibrage statique pour plusieurs scénarii défaillants.

### 3.6.3 Replanification de trajectoires TAEM en présence de défauts

Dans cette section, nous appliquons la première méthode de replanification de trajectoires tolérante aux défauts décrite dans la section 2.5.2 au cas d'une mission de rentrée



atmosphérique. Nous utiliserons également la paramétrisation du modèle à trois degrés de liberté du véhicule en temps libre, introduite dans la section 3.3.2.

En utilisant les expressions complètes des coefficients de portance (3.118) et de traînée (3.119) du véhicule, nous commençons par réécrire les expressions analytiques de l'incidence  $\alpha$ , de la contrainte non linéaire égalitaire de trajectoire ainsi que du facteur de charge  $\Gamma$ . En utilisant (3.118) au lieu de  $C_{L_0}(\alpha, M)$  dans l'expression (3.42), on obtient

$$\alpha = \frac{\frac{2m}{\rho SV \cos \mu} \left( \frac{\gamma'}{\lambda} + \frac{g \cos \gamma}{V} \right) - C_{L_{\delta_e}}(M, \delta_e) - C_{L_{\delta_{bf}}}(M, \delta_{bf}) - C_{L_{\delta_{sb}}}(M, \delta_{sb})}{a_1 f_{C_{L_0}}(M)} - \frac{a_0}{a_1}, \quad (3.123)$$

qui ne présente pas de singularité car  $-80^\circ \leq \mu \leq 80^\circ$  le long de la trajectoire TAEM. De plus, les coefficients  $C_{L_{\delta_e}}(M, \delta_e)$ ,  $C_{L_{\delta_{bf}}}(M, \delta_{bf})$  et  $C_{L_{\delta_{sb}}}(M, \delta_{sb})$  sont entièrement approxi-  
més par des réseaux neurones statiques. Le coefficient de portance en configuration lisse  $C_{L_0}(\alpha, M)$  est approximé en utilisant l'approche par séparation de variables présentée dans la section 3.3.1, tel que

$$C_{L_0}(\alpha, M) = (a_0 + a_1 \alpha) f_{C_{L_0}}(M).$$

En outre, l'équation autonome (3.43) se réécrit

$$\Lambda_\tau(x, u, \delta) = \frac{V'}{\lambda} + g \sin \gamma + \frac{\rho SV^2 \left[ C_{D_0}(\alpha, M) + C_{D_{\delta_e}}(\alpha, M, \delta_e) + C_{D_{\delta_{sb}}}(\alpha, M, \delta_{sb}) \right]}{2m} = 0, \quad (3.124)$$

où le coefficient de traînée total est défini par (3.119).

Finalement, l'expression de la contrainte de facteur de charge (3.14) le long de la trajectoire est modifiée afin de tenir compte des coefficients de force totaux (3.118) et (3.119), telle que

$$\Gamma = \frac{\rho SV^2 \sqrt{C_{L_{tot}}^2(\alpha, M, \delta) + C_{D_{tot}}^2(\alpha, M, \delta)}}{2mg}.$$

En tenant compte des contraintes d'équilibrage statique le long de la trajectoire TAEM, le problème de commande optimale (3.44) peut alors être mis sous la forme générique

(2.52), tel que

$$\text{trouver } (\bar{z}(t), \lambda, \delta(t)) \tag{3.125}$$

t.q.

$$\begin{aligned} & \psi_x(\bar{z}(\tau_0), \lambda) = x_0, \\ & \psi_u(\bar{z}(\tau_0), \lambda, \delta(\tau_0)) = u_0, \\ & \Lambda_\tau(\psi_x(\bar{z}(\tau), \lambda), \psi_u(\bar{z}(\tau), \lambda, \delta(\tau))) = 0, & \tau \in [\tau_0, \tau_f], \\ & C_{m_{tot}}(\psi_x(\bar{z}(\tau), \lambda), \psi_u(\bar{z}(\tau), \lambda, \delta(\tau))) = 0, & \tau \in [\tau_0, \tau_f], \\ & T_{(l,n)_\delta}(\delta(\tau)) = 0, & \tau \in [\tau_0, \tau_f], \\ & 0 \leq \Gamma(\psi_x(\bar{z}(\tau), \lambda), \psi_u(\bar{z}(\tau), \lambda, \delta(\tau))) \leq \Gamma_{\max}, & \tau \in [\tau_0, \tau_f], \\ & 0 \leq \bar{q}(\psi_x(\bar{z}(\tau), \lambda)) \leq \bar{q}_{\max}, & \tau \in [\tau_0, \tau_f], \\ & u_{\min} \leq \psi_u(\bar{z}(\tau), \lambda, \delta(\tau)) \leq u_{\max}, & \tau \in [\tau_0, \tau_f], \\ & \psi_x(\bar{z}(\tau_f), \lambda) = x_f, \\ & \psi_u(\bar{z}(\tau_f), \lambda, \delta(\tau_f)) = u_f, \end{aligned}$$

où le vecteur  $\bar{z}$  est défini par  $\bar{z} = (z_1, z_2, z_3, \dot{z}_1, \dot{z}_2, \dot{z}_3, \ddot{z}_1, \ddot{z}_2, \ddot{z}_3)$  dans l'espace des jets d'ordre infini, et où le temps normalisé  $\tau$  vérifie  $\tau_0 = 0$  et  $\tau_f = 1$ . De plus, le coefficient de moment de tangage total est défini par (3.116) et la contrainte  $T_{(l,n)_\delta}$  associée à l'équilibrage statique latéral du véhicule est donnée par (3.122).

Afin de transformer le problème de commande optimale (3.125) en un problème de programmation non linéaire, nous avons paramétré les sorties plates  $z_1$ ,  $z_2$  et  $z_3$  par des B-splines d'ordre 6 et de multiplicité 3, comportant chacune 6 intervalles. De plus, les gouvernes  $\delta$  sont paramétrées par des B-splines d'ordre 2 et de multiplicité 1, comportant chacune 25 intervalles. Finalement, le scalaire  $\lambda$  associé à la durée finale de la trajectoire est paramétré par une B-spline d'ordre 1, de multiplicité 0 et comportant 1 intervalle. Les contraintes de trajectoires de l'OCP (3.125) sont vérifiées en 25 points de collocation uniformément répartis entre  $\tau_0$  et  $\tau_f$ .

Les conditions cinématiques initiales et finales retenues sont les mêmes que dans la section 3.3.2. De plus, les limites de braquage de gouvernes sont indiquées dans le tableau 3.1. La situation défaillante considérée est un blocage multiple de gouvernes : l'élevon extérieur gauche est bloqué en position à  $17^\circ$  et l'aérofrein est bloqué à  $0^\circ$ . La carte d'équilibrage statique associée à cette panne est représentée sur la figure 3.48(a). La figure 3.49 illustre les trajectoires TAEM obtenues sans prise en compte et avec prise en compte du défaut. Chaque trajectoire a été générée en environ 2 s avec le solveur NPSOL sous l'environnement Matlab. Cette différence de temps CPU, par rapport à ceux obtenus dans la section 3.3.2, est principalement liée à l'augmentation du nombre de variables d'optimisation (4 dans la section 3.3.2, 10 dans le cas présent).

Les figures 3.50(a) et 3.50(b) représentent respectivement les profils de gîte et d'incidence le long de la trajectoire TAEM, avec et sans prise en compte des contraintes d'équilibrage statique.

De plus, la figure 3.51 représente la contrainte non linéaire égalitaire de trajectoire devant

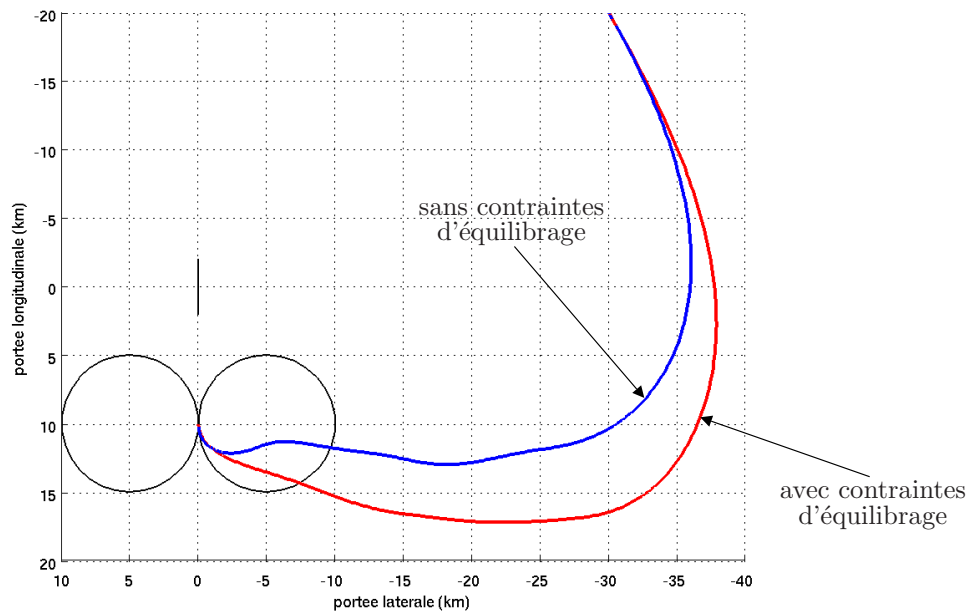
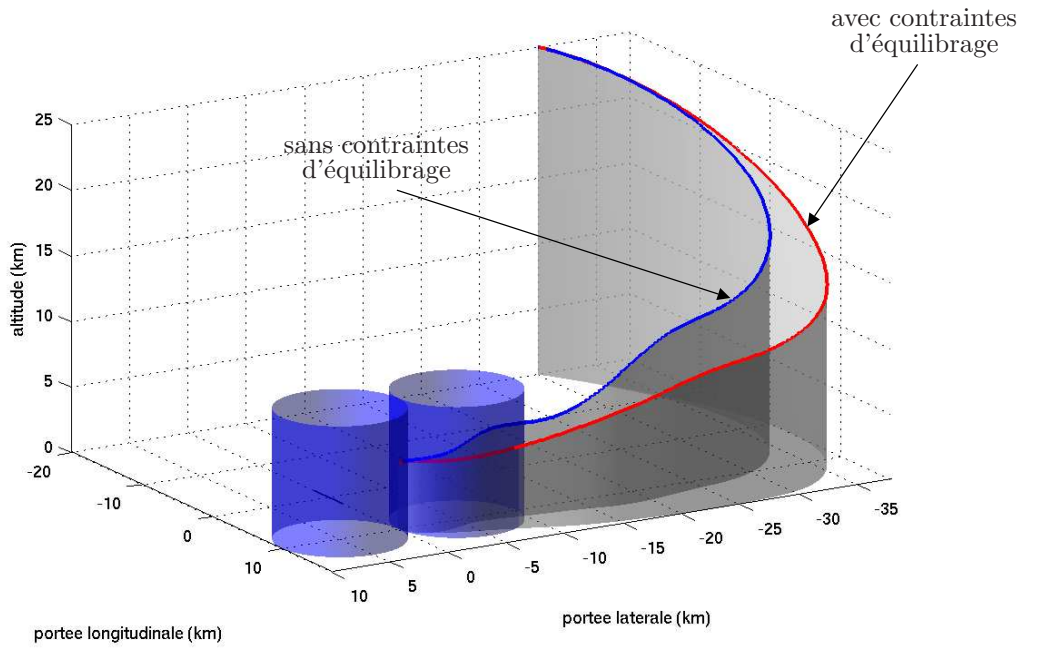


FIG. 3.49 – prise en compte des contraintes d'équilibrage statique pour la planification de trajectoires TAEM.

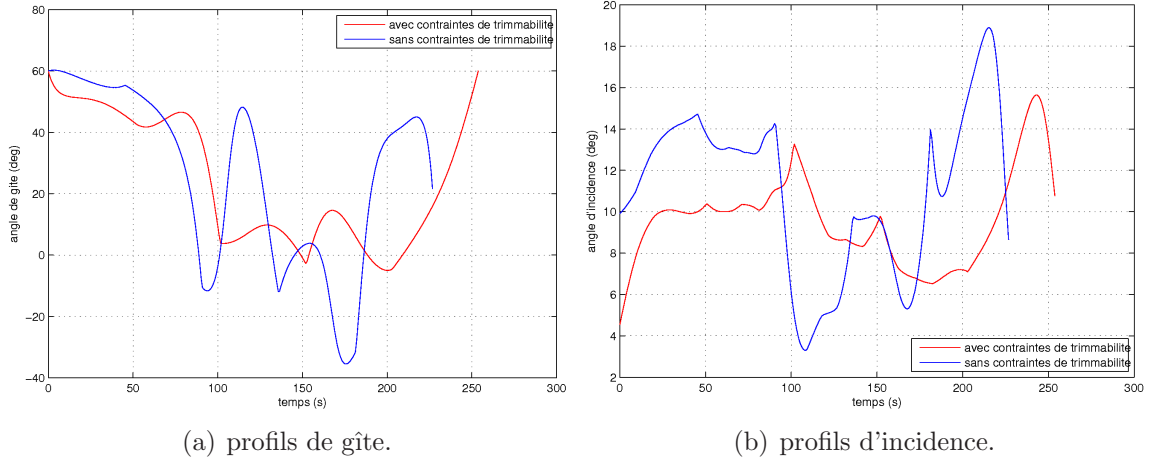


FIG. 3.50 – profils des consignes de guidage sur la trajectoire TAEM de référence.

être identiquement vérifiée aux points de collocation.

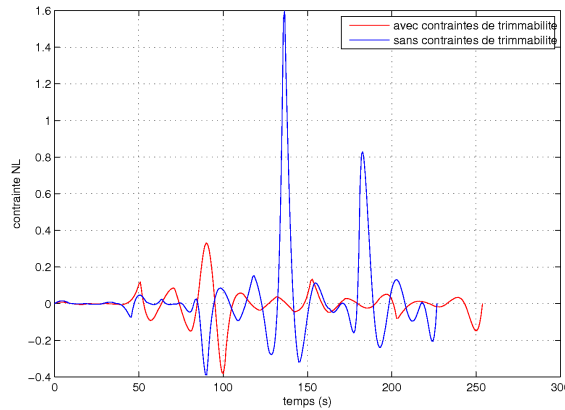


FIG. 3.51 – contrainte égalitaire le long de la trajectoire TAEM ( $m/s^2$ ).

Les contraintes de pression dynamique et de facteur de charge sont représentées sur les figures 3.52(a) et 3.52(b) respectivement. Ces dernières n'excèdent jamais les valeurs maximales admissibles indiquées dans le tableau 3.3.

Lorsque la situation défaillante est prise en compte dans l'algorithme de replanification de trajectoires à bord, les contraintes de trimmabilité statique du véhicule le long de la trajectoire TAEM doivent être vérifiées aux points de collocation. Les figures 3.53(a) et 3.53(b) illustrent la satisfaction de ces contraintes, associées respectivement au mode longitudinal et au mode latéral du véhicule.

Les figures 3.54(a), 3.54(b), 3.54(c) et 3.54(d) représentent les braquages des élévons

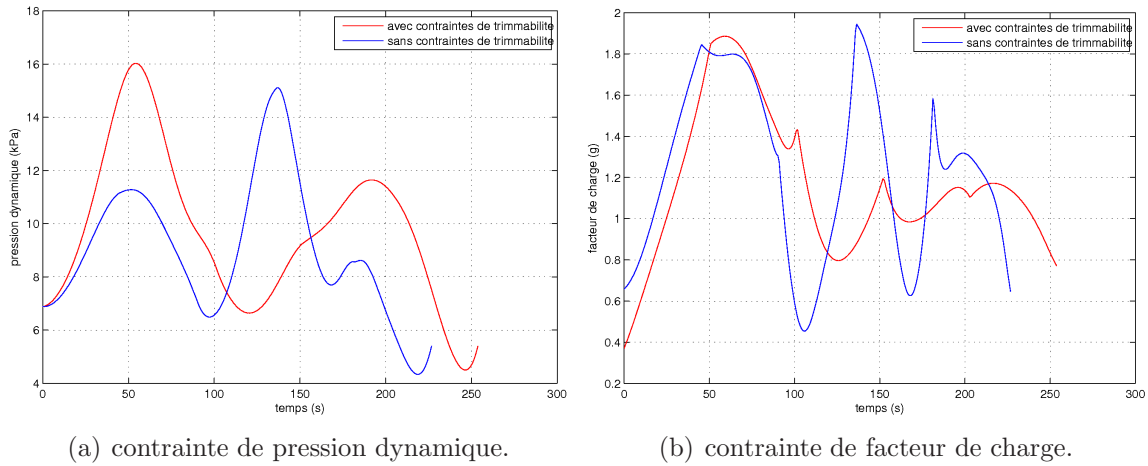


FIG. 3.52 – contraintes mécaniques le long de la trajectoire TAEM de référence.

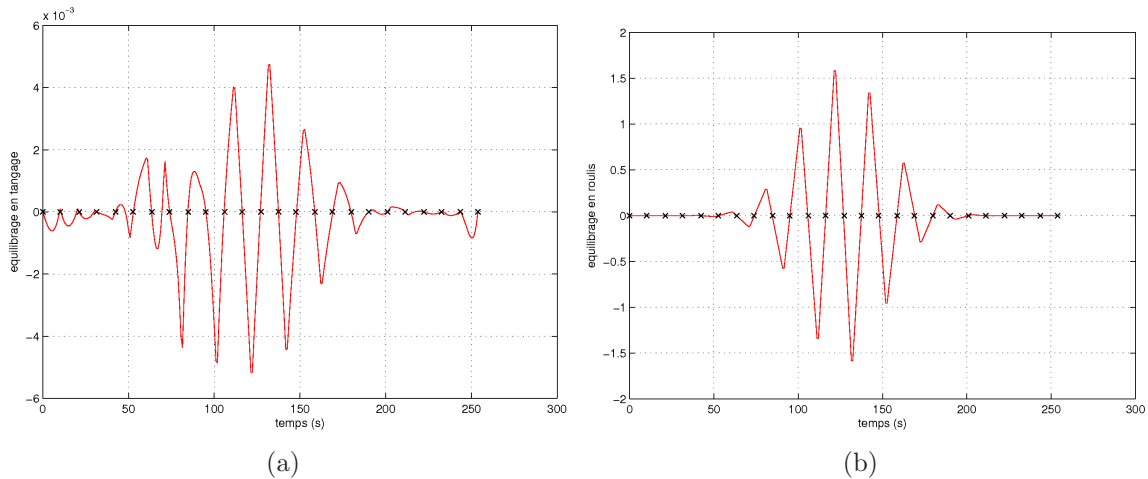


FIG. 3.53 – (a) équilibrage statique en tangage le long de la trajectoire TAEM (b) équilibrage statique en roulis et lacet le long de la trajectoire TAEM.

aux points de collocation.

**Remarque 3.25.** Comme nous l'avons dit précédemment, nous ne cherchons pas ici à résoudre un problème d'allocation de gouvernes, qui est du ressort de la loi de pilotage du véhicule. Ainsi, nous supposons qu'entre les points de collocation, où sont vérifiées les contraintes d'équilibrage statique, le module d'allocation alloue les couples intermédiaires aux différents actionneurs. Cela suppose que l'on se trouve à l'intérieur du domaine d'équilibrage admissible entre les points de collocation, hypothèse que l'on pourra vérifier *a posteriori* grâce à la carte d'équilibrage statique.

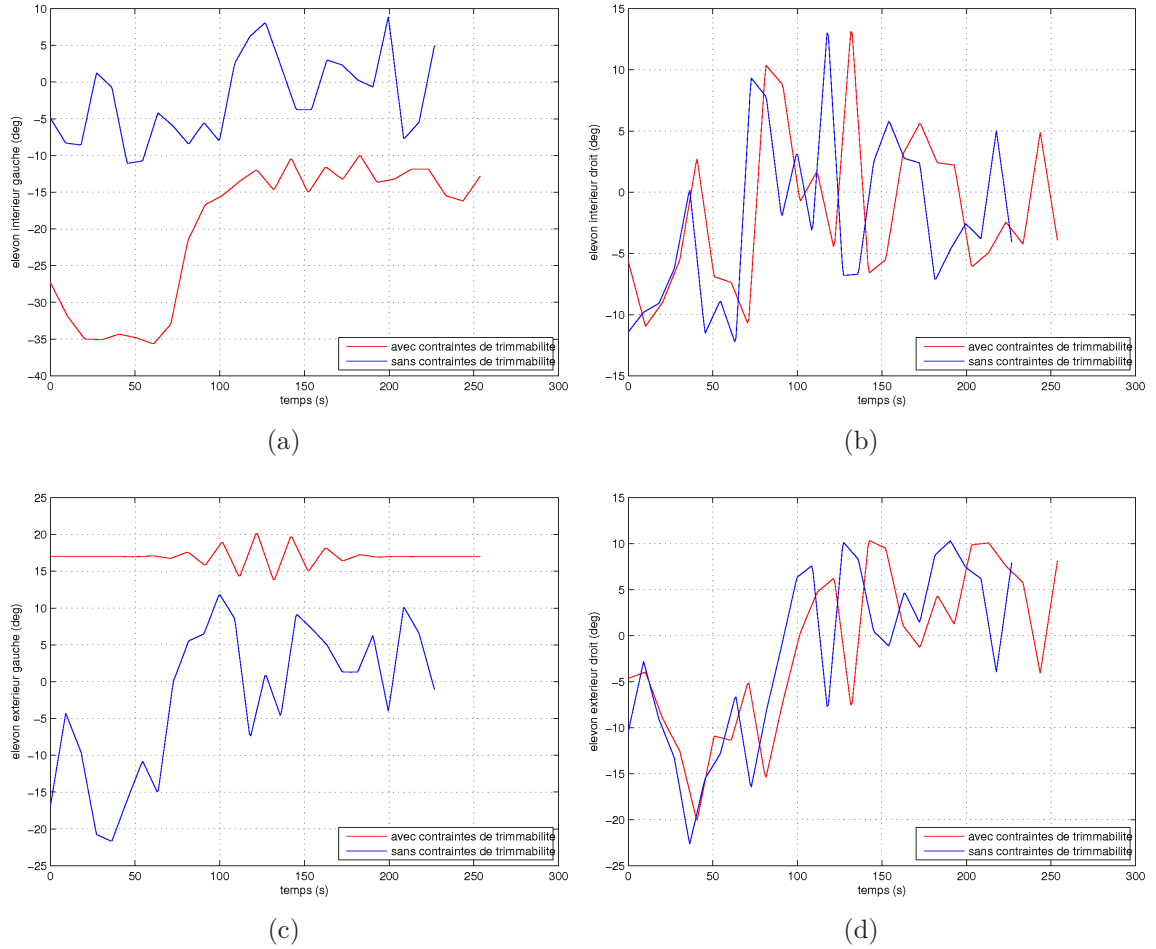


FIG. 3.54 – (a) braquages de l'élevon intérieur gauche  $\delta_{e_{il}}$  (b) braquages de l'élevon intérieur droit  $\delta_{e_{ir}}$  (c) braquages de l'élevon extérieur gauche  $\delta_{e_{ol}}$  (d) braquages de l'élevon extérieur droit  $\delta_{e_{or}}$ .

Les profils de braquage de l'aérofrein et du body flap sont représentés sur les figures 3.55(a) et 3.55(b).

La carte de trimmabilité statique associée à la situation défaillante est représentée sur la figure 3.56(a) en trois dimensions, et projetée dans le plan Mach- $\alpha$  sur la figure 3.56(b). Le bandeau sur la partie droite des figures représente les valeurs de déficience d'équilibrage  $def(i, j)$ , définies dans l'algorithme 12. La trajectoire TAEM initiale, ainsi que la trajectoire tenant compte des contraintes d'équilibrage statique y sont représentées. Clairement, la trajectoire ne tenant pas compte de l'enveloppe de vol réduite du véhicule suite à une panne n'est pas atteignable.

Dans cette section, nous avons augmenté le problème de commande optimale initial afin de tenir compte de situations défaillantes pouvant survenir lors d'une phase critique d'une

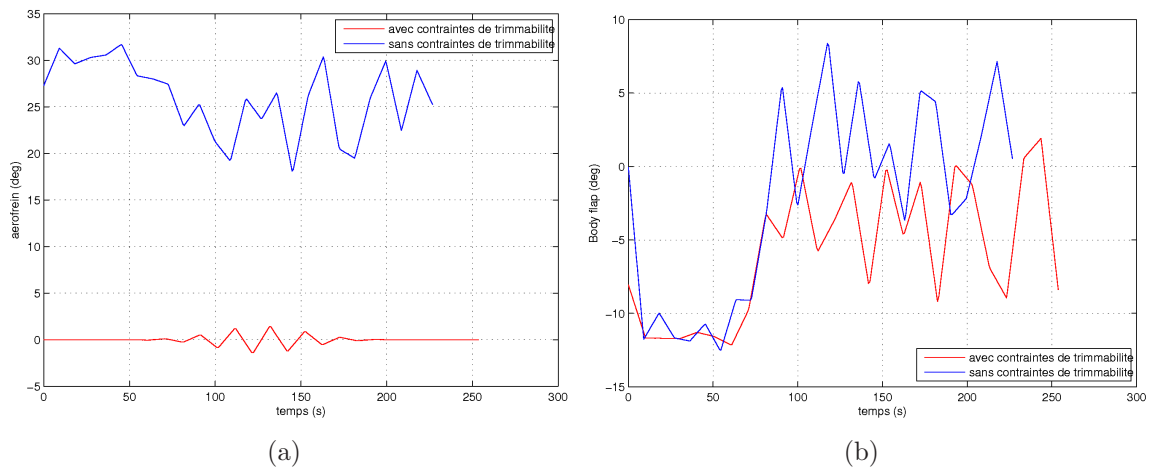


FIG. 3.55 – (a) braquages de l'aérofrein  $\delta_{sb}$  (b) braquages du body flap  $\delta_{bf}$ .

mission de rentrée atmosphérique. Plus précisément, la satisfaction des contraintes d'équilibrage statique en rotation permet de garantir que la nouvelle trajectoire TAEM générée appartient à l'enveloppe de vol réduite du véhicule, suite à une défaillance. La méthodologie proposée, ne nécessitant pas une modélisation intensive des effets des pannes sur la dynamique du véhicule, permet de tenir compte d'un grand nombre de situations défaillantes, mais au prix d'un temps de calcul CPU plus important du fait de l'augmentation du nombre de variables de décision dans le problème NLP.

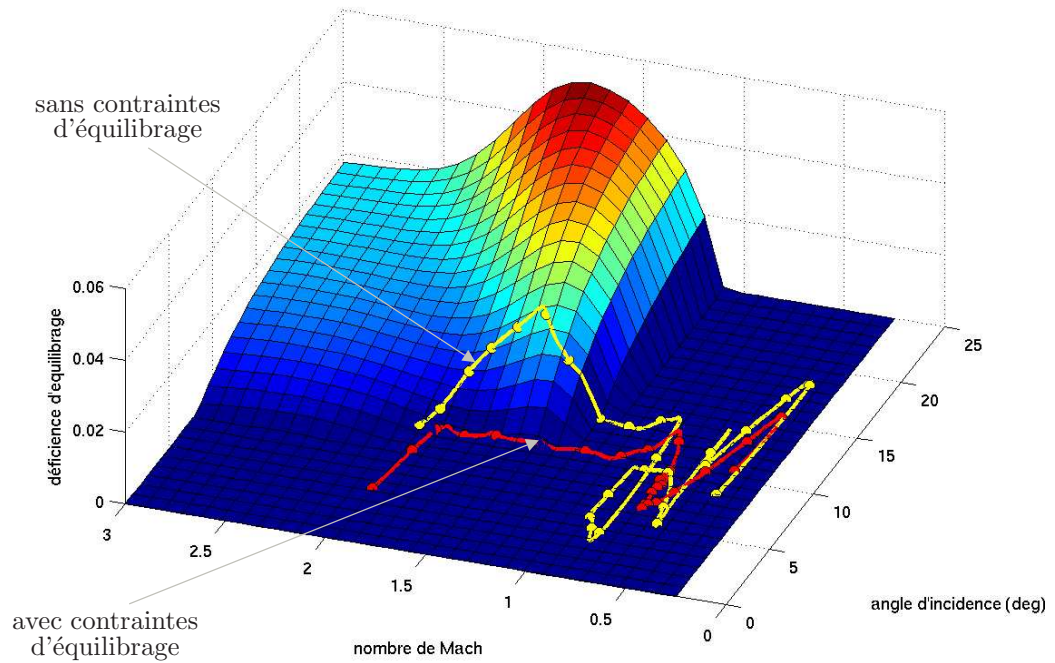
### 3.7 Conclusion

L'objectif de ce chapitre était d'appliquer les techniques de guidage autonome proposées à une mission de rentrée atmosphérique pour la navette spatiale américaine STS-1.

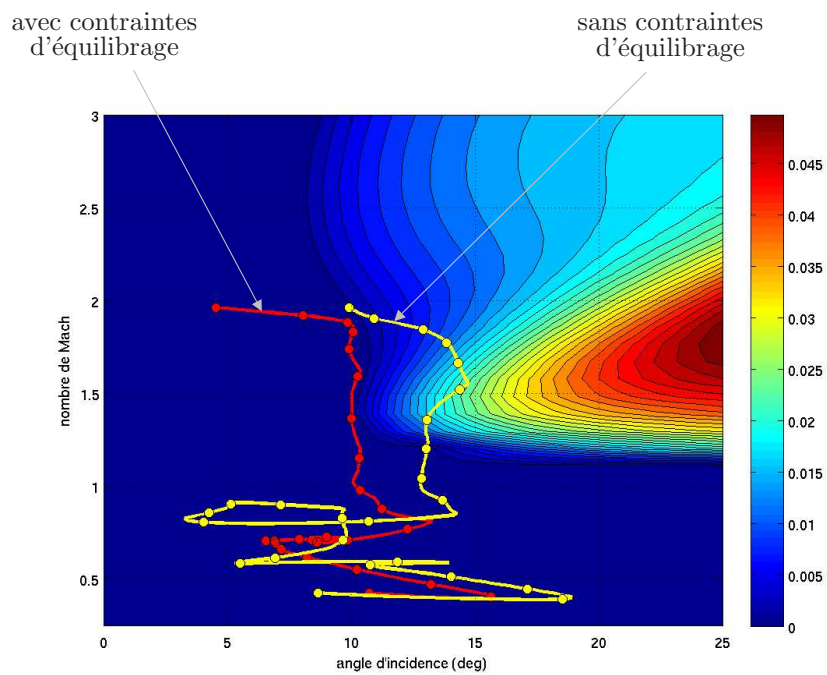
Dans un premier temps, la problématique associée au guidage pour les phases de dissipation d'énergie en zone terminale (TAEM), et d'approche et d'atterrissage automatique (A&L) est décrite, ainsi que le véhicule de rentrée considéré dans le cadre de cette étude. Un simulateur complet à six degrés de liberté, incluant une loi de pilotage non linéaire comportant un module d'allocation de gouvernes en ligne tolérant aux défauts, et un modèle de performances de navigation, a été développé dans le cadre de cette étude.

Ensuite, des trajectoires TAEM et A&L ont été planifiées en utilisant différents paramètres du modèle à trois degrés de liberté régissant la dynamique du centre de masse du véhicule. Notamment, une campagne de simulations Monte Carlo a permis de mettre en valeur le gain de temps CPU escompté par l'utilisation de la propriété de platitude du modèle.

La section suivante a étudié la prise en compte explicite d'un champ de cisaillement de



(a) illustration en 3D.



(b) illustration en 2D.

FIG. 3.56 – projection de la trajectoire TAEM de référence sur la carte d'équilibre statique.



vent dans l'algorithme de planification de trajectoires. De nouvelles trajectoires TAEM ont été obtenues en un temps réduit en utilisant l'approche par platitude.

La démarche d'annexion superellipsoïdale associée au problème de commande optimale a ensuite été illustrée. Des primitives superquadriques déformables ont été générées dans le domaine admissible des contraintes non linéaires de trajectoire, et les paramètres de dimensionnement de ces dernières ont été optimisés par le biais d'un algorithme génétique. Bien qu'induisant un certain degré de conservatisme, la méthode proposée vise à garantir la convergence du problème de programmation linéaire vers un optimum global.

Finalement, la dernière section s'est focalisée sur la replanification de trajectoires TAEM en présence de défauts sur les surfaces de contrôle aérodynamique de l'engin. Des contraintes d'équilibrage statique, associées à l'enveloppe de vol réduite du véhicule, ont alors été intégrées de manière explicite dans le problème de commande optimale, afin de garantir la faisabilité de la trajectoire replanifiée en présence d'une situation défailante.

# Conclusion générale

## Résumé des contributions

Nous rappelons brièvement ci-après nos contributions dans le cadre de cette thèse avant de proposer quelques directions de recherche pour des développements ultérieurs sur les plans théorique, méthodologique et applicatif.

### Première partie

Dans cette partie, nous avons établi un certain nombre de considérations pratiques sur quelques algorithmes de calcul des sorties plates.

Pour les systèmes linéaires à retards, nous avons proposé un algorithme constructif, basé sur des outils classiques de l'algèbre linéaire polynomiale, permettant de calculer des sorties  $\delta$ -plates candidates.

Nous avons également appliqué l'algorithme de calcul formel associé à la détermination de sorties plates au modèle à trois degrés de liberté d'un véhicule de rentrée atmosphérique, en utilisant deux formulations différentes. Pour ces deux formulations, la propriété de platitude du modèle a été prouvée, et une sortie plate candidate a été déterminée.

### Seconde partie

Dans la seconde partie de ce mémoire, nous avons proposé une architecture de guidage autonome, et utilisé le concept de platitude afin de développer une méthodologie complète de guidage.

En premier lieu, une méthodologie d'annexion superellipsoïdale a été développée afin de convexifier les problèmes de commande optimale intervenant dans le cadre des techniques de planification de trajectoires. La méthodologie proposée est basée sur l'optimisation des paramètres de dimensionnement de formes géométriques déformables par un algorithme génétique. Les formes convexes obtenues sont ensuite intégrées dans l'algorithme de planification de trajectoire afin d'améliorer la propriété de convergence des problèmes de programmation non linéaire. En outre, nous avons proposé un certain nombre de résultats nouveaux portant sur les formes superellipsoïdales de dimension  $n$ . Une librairie de fonctions sous l'environnement Matlab, appelée OCEANS (Optimal Convexification by

Evolutionary Algorithm and Superquadrics) a été spécialement développée.

D'autre part, nous avons proposé un cadre méthodologique général permettant d'obtenir un algorithme de replanification de trajectoires tolérant aux défauts. La prise en compte, dans le problème de commande optimale, de contraintes supplémentaires associées à l'équilibrage statique du véhicule le long de la trajectoire, permet de garantir l'appartenance de la trajectoire replanifiée à l'enveloppe de vol réduite en présence de certains défauts d'actionneurs.

Finalement, une méthode permettant la prise en compte de perturbations aérologiques dans l'algorithme de planification de trajectoires a été proposée. En utilisant la propriété de platitude, la prise en compte de champs de cisaillement de vent a été rendue possible, sans nécessiter l'adjonction de variables d'optimisation supplémentaires dans le problème de commande optimale.

## Perspectives

Les résultats obtenus dans ce mémoire offrent des perspectives intéressantes de développements ultérieurs.

Premièrement, l'application à d'autres classes de systèmes (par exemple les systèmes d'équations aux dérivées partielles, les systèmes non entiers ou hybrides) de l'algorithme formel de calcul de sorties plates proposé, pourrait constituer une extension intéressante de nos travaux sur la théorie des systèmes plats. D'autre part, la détermination de sorties plates candidates pour les systèmes non linéaires sans passer par le calcul d'un modèle variationnel pourrait permettre d'éviter le calcul d'un facteur intégrant de la base de formes différentielles, lorsque celle-ci n'est pas intégrable. Ceci pourrait par exemple être réalisé en utilisant une approche géométrique basée sur la méthode d'équivalence de Cartan.

Sur un plan applicatif, nous pensons que la méthodologie générique de guidage autonome proposée pourrait être facilement appliquée et adaptée à d'autres types de missions et de véhicules (par exemples des missions de surveillance par drones, planification de manoeuvres orbitales pour satellites, guidage autonome de missiles, etc...). De plus, un certain nombre d'axes d'amélioration des techniques proposées peuvent être considérés. Notamment, la prise en compte de l'ensemble des équations dynamiques de pilotage dans l'algorithme de replanification de trajectoires pourrait permettre, en plus de l'équilibrage statique (qui contraint les profils d'incidence à rester à l'intérieur de l'enveloppe de vol), d'imposer des contraintes sur les taux angulaires d'incidence  $\dot{\alpha}$  et de gîte  $\dot{\mu}$ , en présence de défauts d'actionneurs. De plus, la gestion du transitoire entre l'apparition d'un défaut et la prise en compte de la trajectoire replanifiée dans la chaîne GNC semble être un point crucial à étudier, car une durée de vol très faible, en situation défaillante, peut conduire très rapidement à des conditions de vol dégradées. En ce qui concerne la méthode d'annexion superellipsoïdale proposée, d'autres formes géométriques déformables pourraient permettre de bénéficier d'un nombre plus important de degrés de liberté pour

---

l'approximation de contraintes non convexes de trajectoire. Par exemple, les formes hyperquadriques pourraient permettre de réduire le nombre de contraintes du problème de commande optimale en approximant toutes les contraintes de trajectoire par une seule forme géométrique, au lieu de considérer l'intersection de plusieurs formes convexes. En outre, une technique d'optimisation de trajectoires basée sur la résolution d'un problème de programmation semi-définie positive, établie à partir du problème de commande optimale convexe, pourrait être utilisée afin de réduire la sensibilité aux conditions initiales, qui conditionne les performances des solveurs de programmation non linéaire. Finalement, il pourrait être intéressant de coupler nos travaux avec ceux réalisés dans le cadre du deuxième volet du projet SICVER<sup>65</sup>, portant sur le développement de stratégies de diagnostic robuste pour les véhicules de rentrée atmosphérique [171], afin d'obtenir une architecture GNC complètement autonome.

---

<sup>65</sup>Stratégies Innovantes pour la Conduite des Véhicules Expérimentaux de Rentrée.



# Annexe A

## Forme normale de Smith

Cette annexe rappelle le concept de forme de Smith d'une matrice définie sur un anneau principal (PID), et donne un algorithme séquentiel permettant de calculer une telle forme. La décomposition de Smith [185] est un outil de calcul formel qui peut s'avérer être très utile lorsque l'on travaille sur des anneaux principaux. Cette dernière est largement utilisée dans différentes branches des mathématiques et de l'ingénierie.

Soit  $A$  un anneau idéal principal, et soit  $\mathcal{M}_{n,m}(A)$  le module des matrices de taille  $n \times m$  sur  $A$ . Malgré les faibles propriétés algébriques de  $\mathcal{M}_{n,m}(A)$ , certaines matrices, admettant une notion d'inverse, sont appelées *matrices unimodulaires* et forment le groupe linéaire  $\mathcal{U}_n(A)$  des matrices inversibles de taille  $n$  sur  $\mathcal{M}_{n,m}(A)$ .

**Définition A.1** ([186]). *Soit  $A$  un anneau idéal principal,  $\mathcal{M}_{n,m}(A)$  le module des matrices de taille  $n \times m$  sur  $A$ . Le groupe élémentaire  $\mathcal{U}_m(A)$  est le sous-groupe normal<sup>66</sup> du groupe linéaire*

$$GL_m(A) = \{U \in \mathcal{M}_{m,m}(A) \mid \exists V \in \mathcal{M}_{m,m}(A) : UV = VU = I_m\},$$

*des matrices inversibles sur  $\mathcal{M}_{m,m}(A)$ , c'est à dire des matrices unimodulaires.*

**Exemple A.1.** Toute matrice triangulaire de  $\mathcal{M}_{m,m}(A)$  ayant l'une des formes suivantes :

$$\begin{pmatrix} 1 & \star & \cdots & \cdots & \star \\ 0 & 1 & \star & \cdots & \star \\ \vdots & 0 & 1 & \star & \star \\ \vdots & \ddots & 0 & 1 & \star \\ 0 & \cdots & \cdots & 0 & 1 \end{pmatrix}, \quad \begin{pmatrix} 1 & 0 & \cdots & \cdots & 0 \\ \star & 1 & 0 & \cdots & 0 \\ \vdots & \star & 1 & 0 & 0 \\ \vdots & \ddots & \star & 1 & 0 \\ \star & \cdots & \cdots & \star & 1 \end{pmatrix},$$

appartient à  $\mathcal{U}_m(A)$ .

En outre, toute matrice  $M \in \mathcal{M}_{m,m}(A)$  définie sur un anneau principal admet une notion de diagonalisation, appelée *décomposition de Smith*.

---

<sup>66</sup>pour toute matrice inversible  $M \in \mathcal{U}_p(A)$ , on a  $M^{-1}\mathcal{U}_m(A)M = \mathcal{U}_p(A)$ .

**Théorème A.1** (Décomposition de Smith [187, 185]). *Soit  $A$  un anneau idéal principal. Étant donné une matrice polynomiale  $M \in \mathcal{M}_{n,m}(A)$ , il existe des matrices  $V \in \mathcal{U}_n(A)$  et  $U \in \mathcal{U}_m(A)$  telles que*

$$VMU = \begin{cases} (\Delta & 0_{n,m-n}) & \text{si } n < m, \\ \begin{pmatrix} \Delta \\ 0_{n-m,m} \end{pmatrix} & \text{si } n > m, \end{cases} \quad (\text{A.1})$$

est la forme de Smith<sup>67</sup> de  $M$ , où  $\Delta = \text{diag}\{\delta_1, \dots, \delta_\sigma, 0, \dots, 0\}$  est une matrice diagonale de taille  $n \times n$  (resp.  $m \times m$ ) dont les éléments  $(\delta_1, \dots, \delta_\sigma)$  sont tels que  $\delta_i$  est un polynôme non nul pour  $i = 1, \dots, \sigma$ , et est un diviseur de  $\delta_j$  pour tout  $\sigma \geq j \geq i$ .

Le groupe  $\mathcal{U}_m(A)$  des matrices unimodulaires de taille  $m \times m$  admet un nombre fini de générateurs. Ces générateurs correspondent à un certain nombre d'actions élémentaires à gauche et à droite :

- Les *actions à droite* consistent à permuter deux colonnes, à multiplier à droite une colonne par une fonction non nulle ou, pour des  $i$  et  $j$  arbitraires, à ajouter à la  $i$ -ème colonne la  $j$ -ème colonne multipliée à droite par un polynôme arbitraire de  $A$ .
- Les *actions à gauche* consistent, de manière analogue, à permuter deux lignes, à multiplier à gauche une ligne par une fonction non nulle ou, pour des  $i$  et  $j$  arbitraires, à ajouter à la  $i$ -ème ligne la  $j$ -ème ligne multipliée à gauche par un polynôme arbitraire de  $A$ .

Chaque action élémentaire peut être représentée au moyen d'une *matrice élémentaire unimodulaire* de type I, II ou III.

Une *matrice élémentaire de type I* est une matrice carrée de la forme

$$E_I \triangleq \begin{pmatrix} 1 & & \cdots & & 0 \\ & \ddots & & & \\ & & 1 & & \\ \vdots & & & u & \vdots \\ & & & & 1 \\ & & & & \ddots \\ 0 & & \cdots & & 1 \end{pmatrix}, \quad E_I^{-1} = \begin{pmatrix} 1 & & \cdots & & 0 \\ & \ddots & & & \\ & & 1 & & \\ \vdots & & & u^{-1} & \vdots \\ & & & & 1 \\ & & & & \ddots \\ 0 & & \cdots & & 1 \end{pmatrix} \quad (\text{A.2})$$

où  $u \in A$  est un élément inversible à la position  $(i, i)$ .

**Proposition A.1.** *Soit  $E_I \in \mathcal{U}_n(A)$  une matrice élémentaire de type I définie par (A.2), et  $u \in A$  un élément inversible. Alors, pour tout  $P \in \mathcal{M}_{n,p}(A)$  et  $Q \in \mathcal{M}_{m,n}(A)$ , l'action à gauche définie par  $E_I P$  consiste à multiplier à gauche la  $i$ -ème ligne de  $P$  par  $u$ , et l'action à droite  $Q E_I$  consiste à multiplier à droite la  $i$ -ème colonne de  $Q$  par  $u$ .*

Une *matrice élémentaire de type II* est formée par la matrice identité, dans laquelle

---

<sup>67</sup>aussi appelée *forme de Jacobson* dans un cadre non commutatif.





**Algorithme 13** : Décomposition de Smith.

**Entrées** : Une matrice  $M \in \mathcal{M}_{n,m}(A)$

**Sorties** : Des matrices unimodulaires  $V \in \mathcal{U}_n(A)$  et  $U \in \mathcal{U}_m(A)$  dont le produit  $VMU$  génère la forme de Smith de  $M$  définie par (A.1).

**Algorithme** :

- 1 En utilisant des matrices élémentaires de type II, réaliser des opérations d'échange de lignes et de colonnes sur  $M$  afin de mettre l'élément de  $M$  de plus faible degré en position (1,1). Appelons cet élément  $a_{1,1}$ .
- 2 A l'aide de matrices élémentaires de type III, soustraire à la deuxième ligne la première ligne multipliée à gauche par le quotient  $q_{2,1}$  de la division euclidienne à gauche  $a_{2,1} = q_{2,1}a_{1,1} + r_{2,1}$ . Si le reste  $r_{2,1}$  de la division est nul, passer à l'étape suivante. Sinon, permuter la première et la deuxième ligne avec une matrice élémentaire de type II et répéter la procédure jusqu'à ce que le terme  $a_{2,1}$  devienne nul.
- 3 Répéter l'étape 2 avec les autres éléments  $a_{k,1}$ ,  $k = 3, \dots, n$ , de la première colonne.
- 4 Appliquer la même procédure à tous les éléments  $a_{1,k}$ ,  $k = 2, \dots, m$  de la première ligne, excepté le premier élément.
- 5 Retourner à l'étape 2 si des termes non nuls, générés à l'étape 4, apparaissent dans la première colonne. Le degré de l'élément  $a_{1,1}$  va diminuer à chaque cycle, jusqu'à ce que l'on obtienne une matrice de la forme

$$M = \begin{pmatrix} \delta_1 & 0 & \cdots & 0 \\ 0 & & & \\ \vdots & \widetilde{M} & & \\ 0 & & & \end{pmatrix}, \quad (\text{A.5})$$

où  $\delta_1 \in A$  est un polynôme unitaire.

- 6 Si un élément de la matrice  $M$  est de degré inférieur à celui de  $a_{1,1}$ , ajouter la colonne où se trouve cet élément à la première colonne avec une matrice élémentaire de type III, et répéter les étapes 2 à 5. Répéter cette opération jusqu'à ce que la forme (A.5) soit atteinte, avec  $\delta_1$  de degré inférieur ou égal à ceux de chaque élément de  $M$ . Ceci va engendrer une réduction additionnelle du degré de l'élément  $a_{1,1}$ .
- 7 Répéter les étapes 1 à 8 avec la matrice  $\widetilde{M}$ .
- 8 La matrice unimodulaire  $V \in \mathcal{U}_n(A)$  (resp.  $U \in \mathcal{U}_m(A)$ ) est finalement obtenue en faisant le produit de toutes les matrices élémentaires unimodulaires à gauche (resp. à droite).

# Annexe B

## Fonctions polynomiales par morceaux et B-splines

Cette annexe rappelle certaines propriétés fondamentales des polynômes continus par morceaux, et plus spécifiquement des B-splines. Le lecteur intéressé pourra trouver plus de précisions sur le sujet dans [172] et [188].

Quelques notations utiles sont d'abord rappelées. Soit  $l$  le nombre de segments polynomiaux d'ordre  $k$ ,  $\{\xi_i\}_{i=0}^l = \{\xi_0, \dots, \xi_l\}$  une séquence strictement croissante constituée de  $(l + 1)$  points, appelés *noeuds de segmentation*<sup>68</sup>, et  $\{s_i\}_{i=0}^l = \{s_0, \dots, s_l\}$  une suite d'entiers non négatifs permettant de spécifier la *continuité* désirée de la courbe  $y(\xi_i) \in C^{s_i}$  aux noeuds de segmentation  $\xi_i$ .

**Définition B.2.** *L'ensemble des fonctions polynomiales par morceaux d'ordre  $k$ , définies sur la séquence de segmentation  $\{\xi_i\}_{i=0}^l$ , où  $l$  représente le nombre de segments considéré, et où  $y(\xi_i)$  est de classe  $C^{s_i}$ , forme un espace vectoriel  $\mathcal{P}_{l,k,s}$  de dimension  $N = \dim(\mathcal{P}_{l,k,s})$ .*

La *séquence nodale*  $\eta$  est alors construite à partir de la séquence de segmentation  $\{\xi_i\}_{i=0}^l$  et du degré de continuité  $C^{s_i}$  désiré en ces points. La séquence nodale représente les points de la courbe où les contraintes associées au problème d'optimisation considéré seront vérifiées. De manière plus précise, la séquence nodale  $\eta$  contient les noeuds de segmentation avec leur multiplicité respective  $k = m_i + s_i$ , où  $s_i$  désigne le degré de continuité paramétrique désiré pour la courbe  $y(t)$  aux noeuds  $\xi_i$ , et  $k$  représente l'ordre de la base de B-splines. La séquence nodale est alors définie par

$$\eta = (\xi_0^1, \dots, \xi_0^{m_0}, \xi_1^1, \dots, \xi_1^{m_1}, \dots, \xi_l^1, \dots, \xi_l^{m_l}).$$

Dans notre cas, nous considérons un type particulier de séquence nodale, connue sous le nom de séquence nodale attachée, caractérisée par le fait qu'elle est discontinue aux points extrêmes de la courbe, i.e.,  $m_0 = m_l = n$ . De plus, même s'il est possible de spécifier le degré de continuité paramétrique de la courbe  $y(t)$  en chacun des segments la constituant, il semble être plus intéressant de pouvoir spécifier le degré de continuité de l'ensemble de

---

<sup>68</sup>les noeuds de segmentation représentent les points où les différents segments de la courbe sont reliés ensembles. La séquence des noeuds de segmentation est alors appelée *séquence de segmentation*.

la courbe (par exemple,  $y(t) \in C^s$ ). Ceci peut être facilement réalisé en répétant  $k$  fois les noeuds extrêmes de la séquence nodale, possédant alors chacun une multiplicité d'ordre  $k$ , et en répétant  $m$  fois les noeuds internes de la séquence, qui possèdent alors une multiplicité d'ordre  $m$ . La séquence nodale suivante regroupe les règles de construction précédentes :

$$\eta = (\xi_0^1, \dots, \xi_0^k, \xi_1^1, \dots, \xi_1^m, \dots, \xi_{l-1}^1, \dots, \xi_{l-1}^m, \xi_l^1, \dots, \xi_l^k).$$

Finalement, nous avons choisi de répartir uniformément les noeuds sur l'intervalle  $[t_0, t_f]$ . En d'autres termes, chaque terme  $\xi_i$  est calculé à partir de la relation  $\xi_i = t_0 + i\xi_s, i = 0, \dots, l$ , où  $\xi_s = (t_f - t_0)/l$ .

Sous les hypothèses précédentes, une courbe  $y(t) \in \mathcal{P}_{l,k,s}$  s'exprime alors sur la base de B-splines par

$$y(t, c) = \sum_{i=0}^{N-1} c_i \mathcal{B}_{i,k}^{(0)}(t), t \in [t_0, t_f], c_i \in \mathbb{R},$$

où  $\mathcal{B}_{i,k}^{(0)}(t)$  représente la dérivée d'ordre 0 de la  $i^{\text{ème}}$  fonction de la base de B-splines de degré  $k$  construite sur la séquence nodale  $\eta$ , et  $c_i$  est le  $i^{\text{ème}}$  *point de contrôle* associé. Les fonctions de la base de B-splines sont des polynômes de Bernstein d'ordre  $k$ , définis de manière récursive par

$$\begin{aligned} \mathcal{B}_{i,0}^{(0)}(t) &= \begin{cases} 1 & \text{si } t \in [t_i, t_{i+1}], \\ 0 & \text{sinon.} \end{cases} \\ \mathcal{B}_{i,k}^{(0)}(t) &= \frac{t - t_i}{t_{i+k+1} - t_i} \mathcal{B}_{i,k-1}^{(0)}(t) + \frac{t_{i+k} - t}{t_{i+k} - t_{i+1}} \mathcal{B}_{i+1,k-1}^{(0)}(t). \end{aligned}$$

La figure B.1 illustre les points caractéristiques d'une courbe B-spline.

**Remarque B.26.** En utilisant les courbes B-splines, il est également possible de générer des objets de plus grande dimension comme des surfaces ou des volumes. Par exemple, une surface B-spline est alors définie par

$$S(u, v, c) = \sum_{i=0}^M \sum_{j=0}^N c_{ij} \mathcal{B}_{i,k}(u) \mathcal{B}_{j,l}(v), \quad u \in [u_0, u_f], \quad v \in [v_0, v_f], \quad p_{ij} \in \mathbb{R},$$

où les  $c_{ij}$  forment un réseau de points de contrôle de la surface.

Les B-splines possèdent un certain nombre de propriétés qui sont décrites de manière détaillée dans [172] et dans [189]. Nous résumons ici les propriétés intéressantes dans le cadre de notre étude :

1. **Support local :** la fonction  $\mathcal{B}_{i,k}^{(0)}(t)$  de la base de B-splines est nulle en dehors de l'intervalle  $[t_i, t_{i+k}]$ .
2. **Nombre de points de contrôle :** le nombre de points de contrôle, i.e., le nombre de coefficients des B-splines, est donné par

$$P = lk - \sum_{i=0}^{l-1} s_i,$$

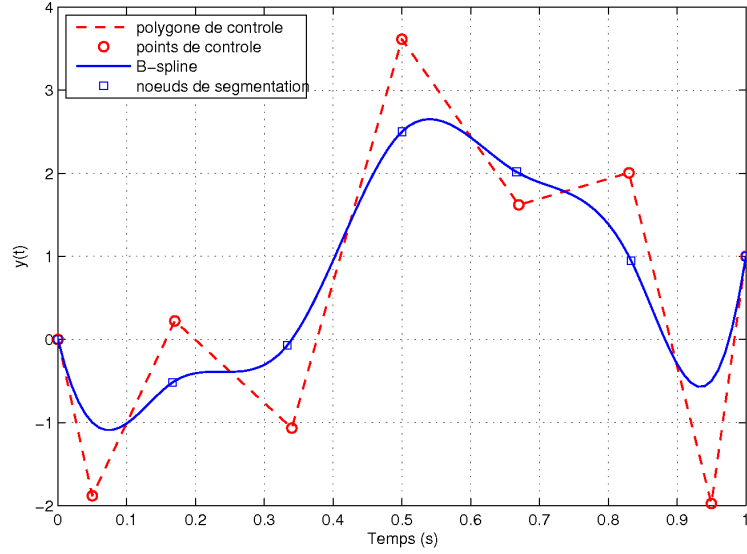


FIG. B.1 – B-spline  $y(t) \in \mathcal{P}(6, 4, 3)$  comportant 6 intervalles ( $l = 6$ ), d'ordre 4 ( $k = 4$ ), et de multiplicité 3 ( $s = 3$ ). Les neuf points de contrôle représentent alors les variables de décision.

où  $l$  est le nombre d'intervalles,  $k$  l'ordre des segments, et  $s$  le degré de continuité. Si tous les noeuds possèdent la même continuité  $s_i = s$ , alors le nombre de points de contrôle est  $P = l(k - s) + s$ .

3. **Enveloppe convexe** : Si  $t \in [t_i, t_{i+1})$ , alors  $y(t)$  se trouve à l'intérieur de l'enveloppe convexe définie par les points de contrôle  $c_{i-k-1}, \dots, c_i$ .
4. **Non négativité** :  $\mathcal{B}_{i,k}^{(0)}(t) \geq 0$  pour tout  $k, i$ , et pour tout  $t$ .
5. **Différentiabilité** : toutes les dérivées des courbes B-splines peuvent être définies sur la séquence de noeuds. La courbe étant  $C^{s_i-1}$  fois continument différentiable aux noeuds, la dérivée de la B-spline est alors définie par

$$y^{(j)}(t, c) = \sum_{i=0}^{N-1} c_i \mathcal{B}_{i,k}^{(j)}(t), t \in [t_0, t_f], c_i \in \mathbb{R},$$

où la  $j^{\text{ème}}$  dérivée temporelle de la fonction  $\mathcal{B}_{i,k}^{(0)}(t)$  de la base de B-splines vaut

$$\mathcal{B}_{i,k}^{(j)}(t) = \frac{k-1}{k-i-j} \left( \frac{t-t_i}{t_{i+k+1}-t_i} \mathcal{B}_{i,k-1}^{(j)}(t) + \frac{t_{i+k}-t}{t_{i+k}-t_{i+1}} \mathcal{B}_{i+1,k-1}^{(j)}(t) \right).$$



# Annexe C

## Équations dynamiques associées à un véhicule de rentrée atmosphérique

Cette annexe rappelle les équations régissant la dynamique à six degrés de liberté d'un véhicule de rentrée atmosphérique. Le lecteur trouvera plus de détails sur le sujet dans [190, 191, 184].

### C.1 Trièdres de référence

#### C.1.1 Trièdres inertiel et géocentrique

Le *trièdre inertiel* terrestre, considéré comme galiléen,  $[R_i] = (O, \vec{X}_i, \vec{Y}_i, \vec{Z}_i)$ , a pour origine le centre de masse de la Terre, et ses axes sont orientés vers trois étoiles suffisamment éloignées pour être considérées comme fixes. L'axe  $\vec{Z}_i$  est alors confondu avec l'axe de rotation de terrestre. Le *trièdre géocentrique*  $[R_g] = (O, \vec{X}_g, \vec{Y}_g, \vec{Z}_g)$ , qui a également pour origine le centre de masse de la Terre, tourne à la *vitesse de rotation terrestre*<sup>69</sup>  $\Omega$  (voir figure C.1).

La matrice de changement de repère  $R_{i/g}$ , permettant de passer du repère inertiel  $[R_i]$  au repère géocentrique  $[R_g]$ , est définie par

$$R_{i/g} = \begin{pmatrix} \cos(\Omega\Delta t) & \sin(\Omega\Delta t) & 0 \\ -\sin(\Omega\Delta t) & \cos(\Omega\Delta t) & 0 \\ 0 & 0 & 1 \end{pmatrix},$$

où  $\Delta t$  représente le *temps sidéral local*, c'est à dire la durée qui s'est écoulée depuis le dernier alignement des deux repères.

#### C.1.2 Trièdre pointant vers le véhicule

Le *trièdre pointant vers le véhicule*  $[R_p] = (O, \vec{X}_p, \vec{Y}_p, \vec{Z}_p)$  a pour origine le centre de masse de la Terre. Le module du *rayon-vecteur*  $|\vec{R}|$  représente la distance séparant

---

<sup>69</sup>qui vaut environ  $7.292115 \cdot 10^{-5}$  rad/s.

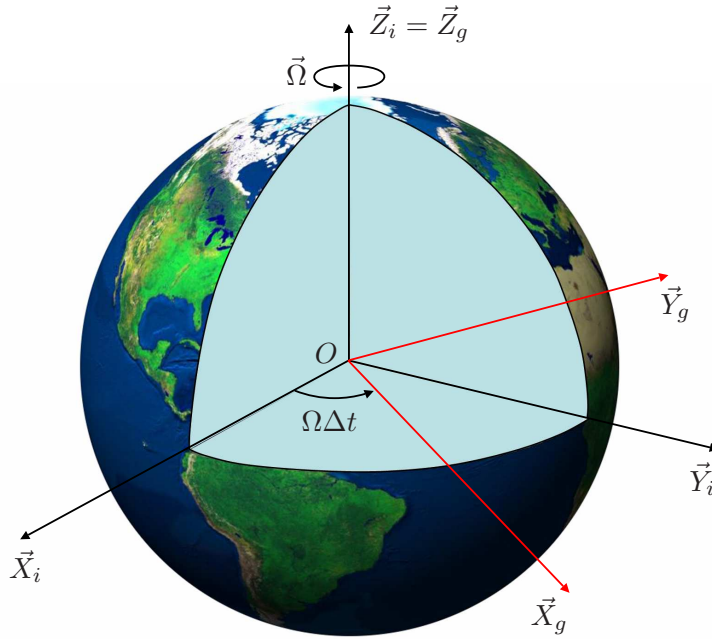


FIG. C.1 – trièdres inertiel et géocentrique.

le centre de gravité  $G$  du véhicule et le centre de masse de la Terre  $O$  (cf figure C.2). L'angle  $\lambda$  désigne la *longitude*, comptée positivement vers l'Est à partir du méridien de Greenwich. La *latitude*  $\varphi$  est comptée positivement vers le Nord à partir de l'équateur. Le trièdre géocentrique  $[R_g]$  et le trièdre pointant vers le véhicule  $[R_p]$  sont liés par la matrice de passage  $R_{g/p}$ , définie par

$$R_{g/p} = \begin{pmatrix} \cos \varphi \cos \lambda & \cos \varphi \sin \lambda & \sin \varphi \\ -\sin \lambda & \cos \lambda & 0 \\ -\sin \varphi \cos \lambda & -\sin \varphi \sin \lambda & \cos \varphi \end{pmatrix}.$$

### C.1.3 Trièdre cinématique

Le *trièdre cinématique*  $[R_c] = (G, \vec{X}_c, \vec{Y}_c, \vec{Z}_c)$  est lié au véhicule et a pour origine son centre de masse  $G$ . Le vecteur vitesse  $\vec{V}$  est porté par l'axe  $\vec{Z}_c$  (voir figure C.3). L'angle formé entre l'horizontale locale au niveau du véhicule et le vecteur vitesse représente la *pente sur la trajectoire*, notée  $\gamma$ . De plus, le *cap* du véhicule, compté positivement par rapport au Nord, correspond à l'angle  $\chi$  existant entre les axes  $\vec{Y}_c$  et  $\vec{Y}_p$ .

On pourra en outre facilement vérifier que la matrice de passage  $R_{p/c}$  du trièdre pointant vers le véhicule  $[R_p]$  au trièdre cinématique  $[R_c]$ , est définie par

$$R_{p/c} = \begin{pmatrix} \cos \gamma & -\sin \gamma \sin \chi & -\sin \gamma \cos \chi \\ 0 & \cos \chi & -\sin \chi \\ \sin \gamma & \cos \gamma \sin \chi & \cos \gamma \cos \chi \end{pmatrix}.$$

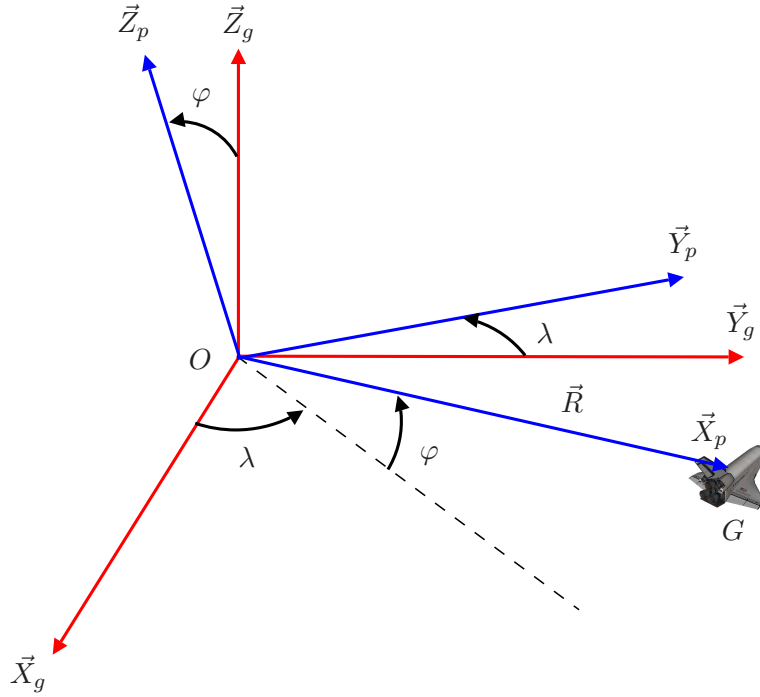


FIG. C.2 – trièdre pointant vers le véhicule.

### C.1.4 Trièdre aérodynamique

Le *trièdre aérodynamique*  $[R_a] = (G, \vec{X}_a, \vec{Y}_a, \vec{Z}_a)$  a pour origine le centre de gravité  $G$  du véhicule. Il est déduit du trièdre cinématique  $[R_c]$  en effectuant une rotation d'amplitude  $\mu$  autour du vitesse  $\vec{V}$ , porté par l'axe  $\vec{Z}_a$  (cf figure C.4). L'angle  $\mu$  est appelé *angle de gîte* (aérodynamique). La *force de portance*  $\vec{L}$  est portée par l'axe  $\vec{X}_a$ , la *force de traînée* par l'axe  $\vec{Z}_a$ , de direction opposée au vecteur vitesse, et la *force latérale*  $\vec{Y}$  est dirigée suivant  $\vec{Y}_a$ .

La matrice de passage  $R_{c/a}$  du trièdre cinématique  $[R_c]$  au trièdre aérodynamique  $[R_a]$  est définie par

$$R_{c/a} = \begin{pmatrix} \cos \mu & \sin \mu & 0 \\ -\sin \mu & \cos \mu & 0 \\ 0 & 0 & 1 \end{pmatrix}.$$

### C.1.5 Trièdre stabilité

Le *trièdre stabilité*  $[R_{st}] = (G, \vec{X}_{st}, \vec{Y}_{st}, \vec{Z}_{st})$  est un trièdre intermédiaire entre le trièdre aérodynamique et le trièdre véhicule, obtenu en projetant le vecteur vitesse  $\vec{V}$  dans le plan de symétrie de l'engin. Il a pour origine le centre de gravité  $G$  du véhicule, et peut être déduit du trièdre aérodynamique  $[R_a]$  par une rotation  $\beta$ , appelée *angle de dérapage*, autour de l'axe  $\vec{X}_a$  (voir figure C.5).



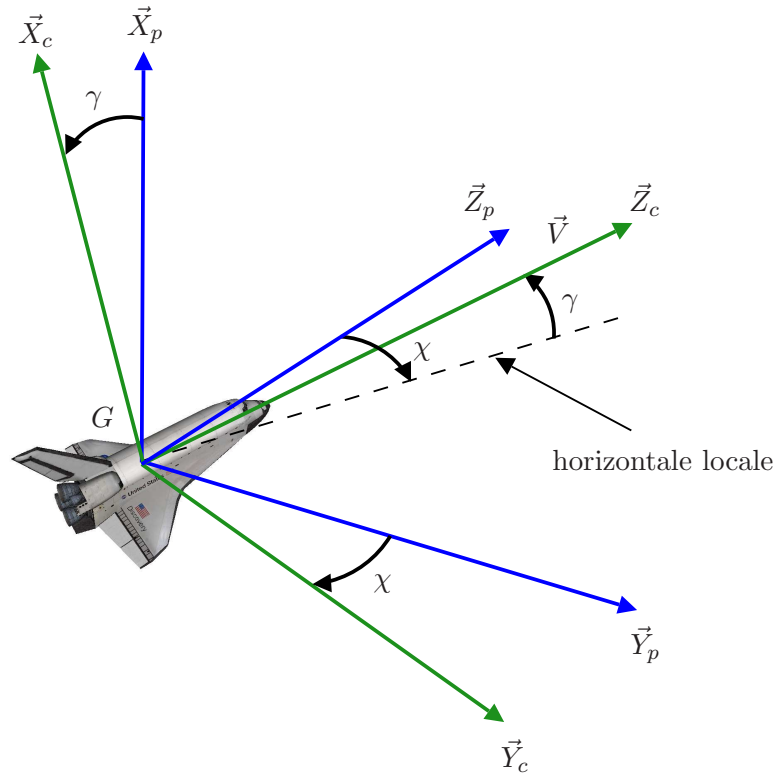


FIG. C.3 – trièdre cinématique.

La matrice de passage  $R_{a/st}$  du trièdre aérodynamique  $[R_a]$  au trièdre stabilité  $[R_{st}]$  est définie par

$$R_{a/st} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos \beta & \sin \beta \\ 0 & -\sin \beta & \cos \beta \end{pmatrix}.$$

### C.1.6 Trièdre véhicule

Le *trièdre véhicule*  $[R_b] = (G, \vec{X}_b, \vec{Y}_b, \vec{Z}_b)$  est lié à la structure de l'engin et a pour origine son centre de gravité  $G$ . L'axe  $\vec{Z}_b$  pointe vers le nez du véhicule et appartient à son plan de symétrie. L'axe  $\vec{Y}_b$  est perpendiculaire au plan de symétrie du véhicule et orienté vers l'aile droite. Enfin, l'axe  $\vec{X}_b$  complète le trièdre de telle sorte que ce dernier soit direct. Le trièdre véhicule se déduit du trièdre stabilité par une rotation  $\alpha$ , appelée *angle d'incidence*, autour de l'axe  $\vec{Y}_{st}$  (voir figure C.6).

La matrice  $R_{st/b}$  permettant de passer du trièdre stabilité  $[R_{st}]$  au trièdre véhicule  $[R_b]$  est définie de la manière suivante :

$$R_{st/b} = \begin{pmatrix} \cos \alpha & 0 & \sin \alpha \\ 0 & 1 & 0 \\ -\sin \alpha & 0 & \cos \alpha \end{pmatrix}.$$

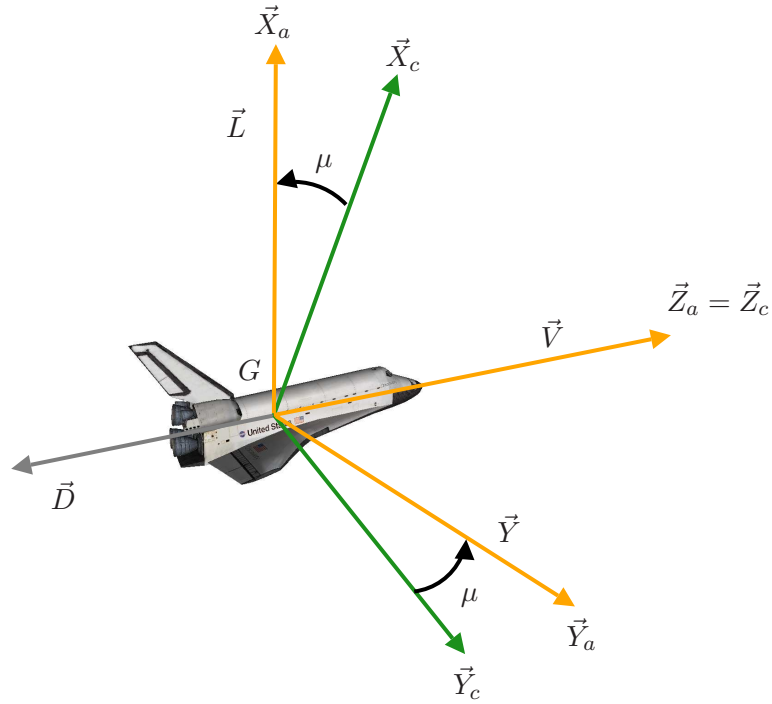


FIG. C.4 – trièdre aérodynamique.

## C.2 Équations dynamiques du véhicule

### C.2.1 Équations de position

La position du centre de masse du véhicule dans l'espace est définie grâce à ses coordonnées géocentriques : le module du rayon-vecteur  $R$ , la longitude  $\lambda$  et la latitude  $\varphi$ . Afin d'établir les équations régissant la dynamique de position du véhicule de rentrée<sup>70</sup>, nous nous plaçons dans le trièdre pointant vers le véhicule  $[R_p]$ . Dans ce trièdre, l'expression du vecteur vitesse aérodynamique par rapport au trièdre géocentrique  $[R_g]$  est donnée par

$$\begin{aligned} \vec{V}|_{R_g}^{[R_p]} &= R_{c/p} \cdot \vec{V}|_{R_g}^{[R_c]} = (R_{p/c})^{-1} \cdot \vec{V}|_{R_g}^{[R_c]} \\ &= (R_{p/c})^T \cdot \begin{bmatrix} 0 \\ 0 \\ V \end{bmatrix} = \begin{bmatrix} V \sin \gamma \\ V \sin \chi \cos \gamma \\ V \cos \chi \cos \gamma \end{bmatrix} \end{aligned} \quad (\text{C.1})$$

<sup>70</sup>couramment appelées *équations de navigation*.

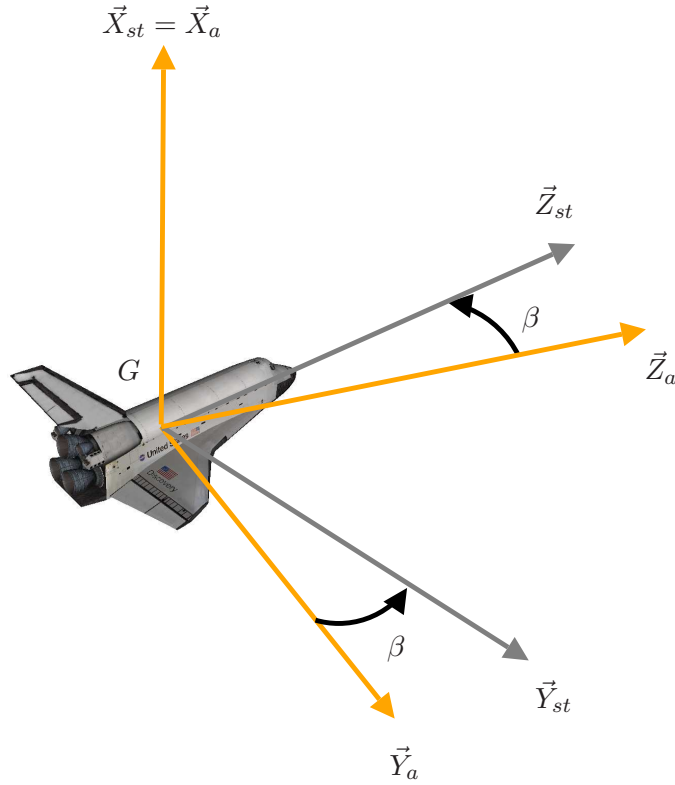


FIG. C.5 – trièdre stabilité.

De plus, la dérivée temporelle de la position du véhicule par rapport au trièdre géocentrique  $[R_g]$ , exprimée dans le trièdre pointant vers le véhicule  $[R_p]$ , est donnée par

$$\begin{aligned} \left. \frac{d(\vec{R})}{dt} \right|_{R_g}^{[R_p]} &= \left. \frac{d(\vec{R})}{dt} \right|_{R_p}^{[R_p]} + \vec{\omega}|_{R_p/R_g}^{[R_p]} \wedge \vec{R}|_{R_p}^{[R_p]} \\ &= \begin{bmatrix} \dot{R} \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} \dot{\lambda} \sin \varphi \\ -\dot{\varphi} \\ \dot{\lambda} \cos \varphi \end{bmatrix} \wedge \begin{bmatrix} R \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} \dot{R} \\ \dot{\lambda} R \cos \varphi \\ R \dot{\varphi} \end{bmatrix} \end{aligned} \quad (\text{C.2})$$

On déduit alors des équations (C.1) et (C.2) les trois équations différentielles régissant la position du véhicule, définies par

$$\dot{R} = V \sin \gamma, \quad (\text{C.3})$$

$$\dot{\varphi} = \frac{V \cos \chi \cos \gamma}{R}, \quad (\text{C.4})$$

$$\dot{\lambda} = \frac{V \sin \chi \cos \gamma}{R \cos \varphi}. \quad (\text{C.5})$$

**Remarque C.27.** L’hypothèse “Terre plate” est largement utilisée lors des phases finales d’une mission de rentrée atmosphérique, notamment lors des phases de dissipation d’énergie en zone terminale (TAEM) et d’approche & atterrissage automatique (A&L). La durée

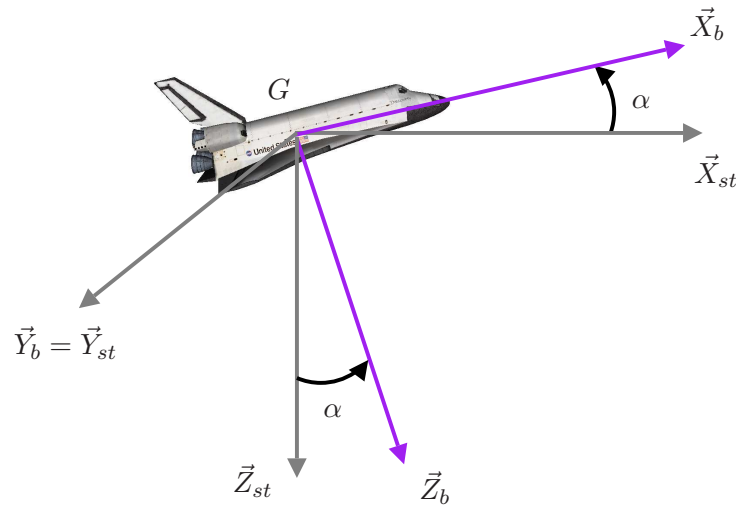


FIG. C.6 – trièdre véhicule.

relativement courte des segments de trajectoire permet alors de négliger le mouvement de rotation terrestre et, par conséquent, les effets de la force de Coriolis et de la force centrifuge sur la dynamique du véhicule. En effectuant le changement de variables

$$\begin{cases} \dot{R} = \dot{h}, \\ \dot{\varphi} = \frac{\dot{x}}{R}, \\ \dot{\lambda} = \frac{\dot{y}}{R \cos \varphi}, \end{cases}$$

le véhicule peut alors être positionné par rapport à un repère cartésien local lié au sol (souvent attaché à l'axe de piste) et associé à la projection conique conforme de Lambert. Les coordonnées relatives du véhicule par rapport à ce repère sont alors communément appelées coordonnées "Terre plate", faisant référence au système de projection utilisé. On obtient alors le système d'équations de navigation

$$\begin{aligned} \dot{x} &= V \cos \chi \cos \gamma, \\ \dot{y} &= V \sin \chi \cos \gamma, \\ \dot{h} &= V \sin \gamma. \end{aligned}$$

### C.2.2 Équations de vitesse

Les équations de vitesse sont obtenues grâce à la seconde loi de Newton, établissant que la somme des efforts extérieurs agissant sur un corps est égale à la dérivée de sa quantité de mouvement :

$$\sum \vec{F}_{ext} = \frac{d}{dt}(m\vec{V}).$$

Dans le cas d'un véhicule de rentrée atmosphérique, les efforts mis en jeu sont de natures différentes :

- le poids  $\vec{P} = m\vec{g}$ ,
- la résultante aérodynamique  $\vec{F}_{aero}$ , composée de la force de portante  $\vec{L}$ , de la force de traînée  $\vec{D}$  et de la force latérale  $\vec{Y}$  (voir figure C.4),
- la force de Coriolis  $\vec{F}_{cor}$  en  $\Omega V$ ,
- la force centrifuge  $\vec{F}_{ctf}$  en  $\Omega^2 R$ ,
- la force d'Euler<sup>71</sup>  $\vec{F}_e$  en  $R \frac{d\Omega}{dt}$ ,
- la force de propulsion  $\vec{F}_{rcs}$  générée par le système de contrôle d'attitude RCS.

On a donc

$$\frac{d(m\vec{V})}{dt} = m\vec{g} + \vec{F}_{aero} + \vec{F}_{rcs} + \vec{F}_{cor} + \vec{F}_{ctf} + \vec{F}_e.$$

Dans la suite, nous allons chercher à exprimer l'ensemble de ces forces dans le trièdre pointant vers le véhicule  $[R_p]$  et par rapport au trièdre géocentrique  $[R_g]$ . Premièrement, on a

$$\begin{aligned} \left. \frac{d(m\vec{V})}{dt} \right|_{R_p} &= \left. \frac{d(m\vec{V})}{dt} \right|_{R_g} + \vec{\omega}|_{R_p/R_g} \wedge \vec{V}|_{R_g} \\ &= m \frac{d}{dt} \begin{bmatrix} V \sin \gamma \\ V \sin \chi \cos \gamma \\ V \cos \chi \cos \gamma \end{bmatrix} + \dot{m} \begin{bmatrix} V \sin \gamma \\ V \sin \chi \cos \gamma \\ V \cos \chi \cos \gamma \end{bmatrix} \\ &\quad + \begin{bmatrix} \dot{\lambda} \sin \varphi \\ -\dot{\varphi} \\ \dot{\lambda} \cos \varphi \end{bmatrix} \wedge \begin{bmatrix} V \sin \gamma \\ V \sin \chi \cos \gamma \\ V \cos \chi \cos \gamma \end{bmatrix}. \end{aligned} \quad (C.6)$$

En remplaçant  $\dot{\lambda}$  et  $\dot{\varphi}$  dans l'équation (C.6) par leurs expressions respectives données par (C.5) et (C.4), il vient :

$$\left. \frac{d(m\vec{V})}{dt} \right|_{R_g} = \begin{bmatrix} \dot{V} \sin \gamma + V \dot{\gamma} \cos \gamma - \frac{V^2 \cos^2 \gamma}{R} + \frac{\dot{m}}{m} V \sin \gamma \\ \dot{V} \cos \gamma \sin \chi - V \dot{\gamma} \sin \gamma \sin \chi + V \dot{\chi} \cos \gamma \cos \chi + \frac{\dot{m}}{m} V \cos \gamma \sin \chi \\ \dots - \frac{V^2 \sin \chi \cos \gamma}{R \cos \varphi} (-\sin \gamma \cos \varphi + \cos \gamma \sin \varphi \cos \chi) \\ \dot{V} \cos \gamma \cos \chi - V \dot{\gamma} \sin \gamma \cos \chi - V \dot{\chi} \cos \gamma \sin \chi + \frac{\dot{m}}{m} V \cos \gamma \cos \chi \\ \dots - \frac{V^2 \cos \gamma}{R \cos \varphi} (-\cos \gamma \sin \varphi + \cos \gamma \sin \varphi \cos^2 \chi - \cos \chi \sin \gamma \cos \varphi) \end{bmatrix} \quad (C.7)$$

De la même manière, les forces extérieures agissant sur le véhicule s'expriment dans le trièdre  $[R_p]$  sous la forme suivante :

<sup>71</sup>la vitesse de rotation terrestre pouvant être considérée comme constante à l'échelle d'une mission de rentrée atmosphérique, nous négligerons la force d'Euler  $\vec{F}_e|_{R_p} = -m \left. \frac{d\vec{\Omega}}{dt} \right|_{R_g} \wedge \vec{R}|_{R_p}$  dans la suite des développements.

- la force de Coriolis

$$\begin{aligned}\vec{F}_{cor}|_{R_p} &= -2m\vec{\Omega}|_{R_p} \wedge \vec{V}|_{R_p} \\ &= \begin{bmatrix} 2m\Omega V \cos \varphi \sin \chi \cos \gamma \\ -2m\Omega V (\cos \varphi \sin \gamma - \sin \varphi \cos \gamma \cos \chi) \\ 2m\Omega V \sin \varphi \sin \chi \cos \gamma \end{bmatrix},\end{aligned}\quad (\text{C.8})$$

où

$$\vec{\Omega}|_{R_p} = R_{g/p} \cdot \vec{\Omega}|_{R_g} = R_{g/p} \cdot \begin{bmatrix} 0 \\ 0 \\ \Omega \end{bmatrix} = \begin{bmatrix} \Omega \sin \varphi \\ 0 \\ \Omega \cos \varphi \end{bmatrix}.$$

- la force centrifuge

$$\begin{aligned}\vec{F}_{ctf}|_{R_p} &= -m\vec{\Omega}|_{R_p} \wedge (\vec{\Omega}|_{R_p} \wedge \vec{R}|_{R_p}) \\ &= \begin{bmatrix} m\Omega^2 R \cos^2 \varphi \\ 0 \\ -m\Omega^2 R \cos \varphi \sin \varphi \end{bmatrix},\end{aligned}\quad (\text{C.9})$$

- la résultante aérodynamique

$$\begin{aligned}\vec{F}_{aero}|_{R_p} &= \begin{bmatrix} L \\ Y \\ -D \end{bmatrix} \cdot R_{a/p} \\ &= \begin{bmatrix} -D \sin \gamma - Y \sin \mu \cos \gamma + L \cos \mu \cos \gamma \\ -D \sin \chi \cos \gamma + Y (\sin \mu \sin \chi \sin \gamma + \cos \mu \cos \chi) \\ \dots - L (\cos \mu \sin \chi \sin \gamma - \sin \mu \cos \chi) \\ -D \cos \chi \cos \gamma + Y (\sin \mu \cos \chi \sin \gamma - \cos \mu \sin \chi) \\ \dots - L (\cos \mu \cos \chi \sin \gamma + \sin \mu \sin \chi) \end{bmatrix},\end{aligned}\quad (\text{C.10})$$

telle que

$$\begin{cases} L = \bar{q} S C_L, \\ Y = \bar{q} S C_Y, \\ D = \bar{q} S C_D, \end{cases}$$

où  $\bar{q} = \frac{1}{2}\rho V^2$  représente la pression dynamique,  $S$  la surface de référence du véhicule et  $C_L$ ,  $C_D$  et  $C_Y$  les coefficients de portance, de traînée et de force latérale, dépendant du nombre de Mach  $M$ , de l'angle d'incidence  $\alpha$ , du dérapage  $\beta$  et des braquages des gouvernes.

- le poids  $\vec{P}|_{R_p} = m\vec{g}|_{R_p}$ , où l'accélération gravitationnelle est portée par la verticale locale (selon l'axe  $\vec{Z}_p$ ), telle que

$$m\vec{g}|_{R_p} = \begin{bmatrix} 0 \\ 0 \\ -mg \end{bmatrix}.\quad (\text{C.11})$$

- la force de propulsion générée par le système de contrôle d'attitude RCS qui est principalement utilisé lors de la phase hypersonique. En ce qui concerne la phase de dissipation d'énergie en zone terminale (TAEM) ainsi que la phase d'approche et d'atterrissage automatique (A&L), nous supposons que  $\vec{F}_{rcs}|_{R_p} = \vec{0}$ .

En combinant les équations (C.7)-(C.11), et après quelques manipulations algébriques élémentaires, nous obtenons les trois équations différentielles régissant la vitesse du véhicule de rentrée :

$$\dot{V} = -\frac{D}{m} - g \sin \gamma - \frac{\dot{m}}{m}V + \Omega^2 R \cos \varphi (\sin \gamma \cos \varphi - \cos \chi \cos \gamma \sin \varphi), \quad (\text{C.12})$$

$$\begin{aligned} \dot{\gamma} &= \frac{1}{mV} (L \cos \mu - Y \sin \mu) + \left( \frac{V}{R} - \frac{g}{V} \right) \cos \gamma + 2\Omega \sin \chi \cos \varphi \\ &+ \frac{\Omega^2 R}{V} \cos \varphi (\cos \chi \sin \gamma \sin \varphi + \cos \gamma \cos \varphi), \end{aligned} \quad (\text{C.13})$$

$$\begin{aligned} \dot{\chi} &= \frac{1}{mV \cos \gamma} (L \sin \mu + Y \cos \mu) + \frac{V}{R} \cos \gamma \sin \chi \tan \varphi + 2\Omega (\sin \varphi - \cos \chi \tan \gamma \cos \varphi) \\ &+ \frac{\Omega^2 R}{V \cos \gamma} \sin \chi \sin \varphi \cos \varphi. \end{aligned} \quad (\text{C.14})$$

**Remarque C.28.** Sous l'hypothèse ‘‘Terre plate’’, la force de Coriolis et la force centrifuge sont négligées. De plus, sous l'hypothèse que le système RCS n'est pas utilisé durant les phases TAEM et A&L, la masse du véhicule ne varie pas, i.e.,  $\dot{m} = 0$ . Finalement, en remarquant que  $V \ll R$ , les équations de vitesse (C.12)-(C.14) peuvent se réécrire

$$\begin{aligned} \dot{V} &= -\frac{D}{m} - g \sin \gamma, \\ \dot{\gamma} &= \frac{1}{mV} (L \cos \mu - Y \sin \mu) - \frac{g}{V} \cos \gamma, \\ \dot{\chi} &= \frac{1}{mV \cos \gamma} (L \sin \mu + Y \cos \mu). \end{aligned}$$

### C.2.3 Équations de vitesses angulaires

Les équations de vitesse angulaire représentent la dynamique de l'angle d'incidence  $\alpha$ , de l'angle de dérapage  $\beta$  ainsi que de l'angle de gîte  $\mu$ . Afin d'établir les équations correspondantes, nous allons chercher à exprimer la vitesse angulaire du trièdre cinématique  $[R_c]$  par rapport au trièdre géocentrique  $[R_g]$ . La façon la plus simple de procéder est de décomposer cette transformation en une somme de trois rotations élémentaires, exprimées dans le trièdre stabilité  $[R_{st}]$ , telles que

$$\vec{\omega}|_{R_c/R_g}^{[R_{st}]} = \vec{\omega}|_{R_c/R_a}^{[R_{st}]} + \vec{\omega}|_{R_a/R_b}^{[R_{st}]} + \vec{\omega}|_{R_b/R_g}^{[R_{st}]}, \quad (\text{C.15})$$

où  $\vec{\omega}|_{R_c/R_g}^{[R_{st}]}$  désigne la vitesse angulaire du trièdre cinématique  $[R_c]$  par rapport au trièdre géocentrique  $[R_g]$ ,  $\vec{\omega}|_{R_c/R_a}^{[R_{st}]}$  la vitesse angulaire du trièdre cinématique  $[R_c]$  par rapport au trièdre aérodynamique  $[R_a]$ ,  $\vec{\omega}|_{R_a/R_b}^{[R_{st}]}$  la vitesse angulaire du trièdre aérodynamique  $[R_a]$

par rapport au trièdre véhicule  $[R_b]$ , et  $\vec{\omega}|_{R_b/R_g}^{[R_{st}]}$  la vitesse angulaire du trièdre véhicule  $[R_b]$  par rapport au trièdre géocentrique  $[R_g]$ .

Les différents vecteurs de rotation s'expriment dans leurs trièdres de définition sous la forme

$$\vec{\omega}|_{R_c/R_g} = \dot{\gamma}\vec{X}_c + \dot{\chi}\vec{Z}_g, \quad (\text{C.16})$$

$$\vec{\omega}|_{R_c/R_a} = -\vec{\omega}|_{R_a/R_c} = -\dot{\mu}\vec{X}_a, \quad (\text{C.17})$$

$$\vec{\omega}|_{R_a/R_b} = -\dot{\alpha}\vec{Y}_{st} + \dot{\beta}\vec{Z}_a, \quad (\text{C.18})$$

$$\vec{\omega}|_{R_b/R_g} = p\vec{X}_b + q\vec{Y}_b + r\vec{Z}_b. \quad (\text{C.19})$$

En effectuant les changements de repères appropriés, les expressions (C.16)-(C.19) s'écrivent dans le trièdre stabilité sous la forme

$$\begin{aligned} \vec{\omega}|_{R_c/R_g}^{[R_{st}]} &= R_{c/a} \cdot R_{a/st} \cdot (\dot{\gamma}\vec{X}_c + \dot{\chi}\vec{Z}_g) \\ &= \begin{bmatrix} \cos \mu & \sin \mu & 0 \\ -\sin \mu & \cos \mu & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \beta & \sin \beta \\ 0 & -\sin \beta & \cos \beta \end{bmatrix} \begin{bmatrix} -\dot{\chi} \cos \gamma \\ \dot{\gamma} \\ -\dot{\chi} \sin \gamma \end{bmatrix} \\ &= \begin{bmatrix} -\dot{\chi} \cos \gamma \cos \mu + \dot{\gamma} \sin \mu \\ \cos \beta (\dot{\chi} \sin \mu \cos \gamma + \dot{\gamma} \cos \mu) - \dot{\chi} \sin \beta \sin \gamma \\ -\sin \beta (\dot{\chi} \sin \mu \cos \gamma + \dot{\gamma} \cos \mu) - \dot{\chi} \cos \beta \sin \gamma \end{bmatrix}, \end{aligned} \quad (\text{C.20})$$

$$\vec{\omega}|_{R_c/R_a}^{[R_{st}]} = R_{a/st} \cdot (-\dot{\mu}\vec{X}_a) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \beta & \sin \beta \\ 0 & -\sin \beta & \cos \beta \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ -\dot{\mu} \end{bmatrix} = \begin{bmatrix} 0 \\ -\dot{\mu} \sin \beta \\ -\dot{\mu} \cos \beta \end{bmatrix} \quad (\text{C.21})$$

$$\begin{aligned} \vec{\omega}|_{R_a/R_b}^{[R_{st}]} &= -\dot{\alpha}\vec{Y}_{st} + \dot{\beta}\vec{Z}_a = -\begin{bmatrix} 0 \\ \dot{\alpha} \\ 0 \end{bmatrix} + \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \beta & \sin \beta \\ 0 & -\sin \beta & \cos \beta \end{bmatrix} \begin{bmatrix} -\dot{\beta} \\ 0 \\ 0 \end{bmatrix} \\ &= \begin{bmatrix} -\dot{\beta} \\ -\dot{\alpha} \\ 0 \end{bmatrix}, \end{aligned} \quad (\text{C.22})$$

$$\begin{aligned} \vec{\omega}|_{R_b/R_g}^{[R_{st}]} &= R_{b/st} \cdot (p\vec{X}_b + q\vec{Y}_b + r\vec{Z}_b) = \begin{bmatrix} \cos \alpha & 0 & \sin \alpha \\ 0 & 1 & 0 \\ -\sin \alpha & 0 & \cos \alpha \end{bmatrix} \begin{bmatrix} -r \\ q \\ p \end{bmatrix} \\ &= \begin{bmatrix} -r \cos \alpha + p \sin \alpha \\ q \\ r \sin \alpha + p \cos \alpha \end{bmatrix}. \end{aligned} \quad (\text{C.23})$$



Finalement, en utilisant la loi de composition (C.15) et les équations (C.20)-(C.23), on obtient le système de trois équations différentielles recherché :

$$\dot{\alpha} = -\tan \beta (p \cos \alpha + r \sin \alpha) + q - \frac{\dot{\chi} \cos \gamma \sin \mu + \dot{\gamma} \cos \mu}{\cos \beta}, \quad (\text{C.24})$$

$$\dot{\beta} = p \sin \alpha - r \cos \alpha + \dot{\chi} \cos \gamma \cos \mu - \dot{\gamma} \sin \mu, \quad (\text{C.25})$$

$$\dot{\mu} = \frac{p \cos \alpha + r \sin \alpha}{\cos \beta} + \dot{\chi} \sin \gamma + \tan \beta (\dot{\chi} \cos \gamma \sin \mu + \dot{\gamma} \cos \mu). \quad (\text{C.26})$$

### C.2.4 Équations d'attitude

Les équations régissant la dynamique des vitesses de roulis  $p$ , de tangage  $q$  et de lacet  $r$ , s'obtiennent en appliquant le théorème du moment cinétique, établissant que la somme des moments extérieurs agissant sur un corps est égale à la dérivée de son moment cinétique, i.e.,

$$\sum \vec{M}_{ext} = \frac{d}{dt}(\vec{H}), \quad (\text{C.27})$$

où  $\vec{H} = J\vec{\omega}$  désigne le moment cinétique,  $\vec{\omega} = [p, q, r]^T$  le vecteur de rotations, et  $J$  la matrice d'inertie (symétrique) associée au véhicule de rentrée, telle que

$$J = \begin{bmatrix} I_{xx} & -I_{xy} & -I_{xz} \\ -I_{xy} & -I_{yy} & -I_{yz} \\ -I_{xz} & -I_{yz} & I_{zz} \end{bmatrix}.$$

Le développement de l'équation (C.27) conduit à

$$\sum \vec{M}_{ext} = \dot{\vec{H}} + \vec{\omega} \times \vec{H} = \dot{J}\vec{\omega} + J\dot{\vec{\omega}} + \vec{\omega} \times \vec{H}, \quad (\text{C.28})$$

où  $\dot{J}$  représente la matrice contenant les dérivées inertielles, telle que

$$\dot{J} = \begin{bmatrix} \dot{I}_{xx} & -\dot{I}_{xy} & -\dot{I}_{xz} \\ -\dot{I}_{xy} & -\dot{I}_{yy} & -\dot{I}_{yz} \\ -\dot{I}_{xz} & -\dot{I}_{yz} & \dot{I}_{zz} \end{bmatrix}.$$

En isolant le vecteur  $\dot{\vec{\omega}} = [\dot{p}, \dot{q}, \dot{r}]^T$  dans l'équation (C.28), il vient

$$\dot{\vec{\omega}} = J^{-1} \left[ -\vec{\omega} \times \vec{H} - \dot{J}\vec{\omega} + \sum \vec{M}_{ext} \right], \quad (\text{C.29})$$

où  $J^{-1}$  représente l'inverse de la matrice d'inertie, telle que

$$J^{-1} = \begin{bmatrix} g_{pl} & g_{pm} & g_{pn} \\ g_{ql} & g_{qm} & g_{qn} \\ g_{rl} & g_{rm} & g_{rn} \end{bmatrix},$$

dont les éléments sont donnés par

$$\begin{aligned}
 g_{pl} &= (I_{yy}I_{zz} - I_{yz}^2)\Delta, \\
 g_{ql} &= (I_{xy}I_{zz} + I_{xz}I_{yz})\Delta, \\
 g_{rl} &= (I_{xy}I_{yz} + I_{xz}I_{yy})\Delta, \\
 g_{pm} &= (I_{xy}I_{zz} + I_{xz}I_{yz})\Delta, \\
 g_{qm} &= (I_{xx}I_{zz} - I_{xz}^2)\Delta, \\
 g_{rm} &= (I_{xx}I_{yz} + I_{xy}I_{xz})\Delta, \\
 g_{pn} &= (I_{xy}I_{yz} + I_{xz}I_{yy})\Delta, \\
 g_{qn} &= (I_{xx}I_{yz} + I_{xz}I_{xy})\Delta, \\
 g_{rn} &= (I_{xx}I_{yy} - I_{xy}^2)\Delta,
 \end{aligned}$$

avec

$$\Delta = \frac{-1}{2I_{xz}I_{xy}I_{yz} - I_{xx}I_{yy}I_{zz} + I_{xx}I_{yz}^2 + I_{yy}I_{xz}^2 + I_{zz}I_{xy}^2}. \quad (\text{C.30})$$

La somme des moments extérieurs est constituée de la somme des moments aérodynamiques et de ceux générés par les tuyères du système RCS, telle que

$$\sum \vec{M}_{ext} = \begin{bmatrix} L_{aero} + L_{rcs} \\ M_{aero} + M_{rcs} \\ N_{aero} + N_{rcs} \end{bmatrix},$$

où les moments propulsifs sont considérés comme étant nuls durant les phases TAEM et A&L, et les moments aérodynamiques sont définis par

$$\begin{cases} L_{aero} = \bar{q}SbC_l \\ M_{aero} = \bar{q}S\bar{c}C_m \\ N_{aero} = \bar{q}SbC_n \end{cases},$$

où  $\bar{q}$  est la pression dynamique,  $b$  l'envergure des ailes,  $\bar{c}$  la longueur de la corde moyenne du profil d'aile. Les termes  $C_l$ ,  $C_m$  et  $C_n$  désignent respectivement les coefficients de moment de roulis, tangage et lacet, dépendant du nombre de Mach  $M$ , de l'incidence  $\alpha$ , du dérapage  $\beta$ , ainsi que des braquages des gouvernes.

On obtient à partir de l'équation (C.29) le système de trois équations différentielles régissant la dynamique des vitesses de roulis, tangage et lacet.

$$\begin{aligned}
 \dot{p} &= I_{ppp}p^2 + I_{pqq}q^2 + I_{prrr}r^2 + I_{ppq}pq + I_{ppr}pr + I_{pqr}qr + \dot{I}_{pp}p + \dot{I}_{pq}q + \dot{I}_{pr}r \\
 &+ g_{pl}(L_{aero} + L_{rcs}) + g_{pm}(M_{aero} + M_{rcs}) + g_{pn}(N_{aero} + N_{rcs}), \quad (\text{C.31})
 \end{aligned}$$

$$\begin{aligned}
 \dot{q} &= I_{qpp}p^2 + I_{qqq}q^2 + I_{qrr}r^2 + I_{qpq}pq + I_{qpr}pr + I_{qqr}qr + \dot{I}_{qp}p + \dot{I}_{qq}q + \dot{I}_{qr}r \\
 &+ g_{ql}(L_{aero} + L_{rcs}) + g_{qm}(M_{aero} + M_{rcs}) + g_{qn}(N_{aero} + N_{rcs}), \quad (\text{C.32})
 \end{aligned}$$

$$\begin{aligned}
 \dot{r} &= I_{rpp}p^2 + I_{rqq}q^2 + I_{rrr}r^2 + I_{rpq}pq + I_{rpr}pr + I_{rqr}qr + \dot{I}_{rp}p + \dot{I}_{rq}q + \dot{I}_{rr}r \\
 &+ g_{rl}(L_{aero} + L_{rcs}) + g_{rm}(M_{aero} + M_{rcs}) + g_{rn}(N_{aero} + N_{rcs}), \quad (\text{C.33})
 \end{aligned}$$

avec

$$\begin{aligned}
 I_{ppp} &= (I_{xy}^2 I_{yz} + I_{xy} I_{xz} I_{yy} - I_{xy} I_{xz} I_{zz} - I_{xz}^2 I_{yz}) \Delta, \\
 I_{pqq} &= (I_{yz} I_{yy} I_{zz} - I_{yz}^2 - I_{xy}^2 I_{yz} - I_{xz} I_{yy} I_{xy}) \Delta, \\
 I_{prr} &= (I_{yz}^2 + I_{xy} I_{zz} I_{xz} + I_{xz}^2 I_{yz} - I_{yz} I_{yy} I_{zz}) \Delta, \\
 I_{ppq} &= (I_{xy} I_{yz} I_{xx} + I_{xz} I_{yy} I_{xx} + I_{xz} I_{yy} I_{zz} - 2I_{xz} I_{yz}^2 - I_{yz} I_{xy} I_{zz} - I_{xy} I_{yz} I_{yy} - I_{xz} I_{yy}^2) \Delta, \\
 I_{pqr} &= (I_{yy}^2 I_{zz} - I_{yy} I_{zz}^2 + I_{yz}^2 I_{zz} - I_{yz}^2 I_{yy} + I_{xy}^2 I_{zz} - I_{xz}^2 I_{yy}) \Delta, \\
 I_{ppr} &= (I_{xy} I_{zz}^2 + I_{xz} I_{yz} I_{zz} + 2I_{xy} I_{yz}^2 + I_{yz} I_{xz} I_{yy} - I_{xy} I_{yy} I_{zz} - I_{xy} I_{zz} I_{xx} - I_{xz} I_{yz} I_{xx}) \Delta,
 \end{aligned}$$

$$\begin{aligned}
 I_{qpp} &= (I_{xz}^3 + I_{xy} I_{yz} I_{xx} - I_{xz} I_{zz} I_{xx} + I_{xy}^2 I_{xz}) \Delta, \\
 I_{qqq} &= (I_{yz} I_{xy} I_{zz} + I_{xz} I_{yz}^2 - I_{xy} I_{yz} I_{xx} - I_{xy}^2 I_{xz}) \Delta, \\
 I_{qrr} &= (I_{xx} I_{xz} I_{zz} - I_{yz} I_{xy} I_{zz} - I_{xz} I_{yz}^2 - I_{xz}^3) \Delta, \\
 I_{ppq} &= (I_{xx}^2 I_{yz} + I_{xy} I_{xz} I_{xx} - I_{yz} I_{zz} I_{xx} - I_{yz} I_{yy} I_{xx} + I_{xy} I_{zz} I_{xz} + 2I_{xz}^2 I_{yz} - I_{xz} I_{yy} I_{xy}) \Delta, \\
 I_{qqr} &= (I_{yz} I_{xz} I_{yy} + I_{xy} I_{zz} I_{xx} - I_{xz} I_{yz} I_{xx} - I_{xz} I_{yz} I_{zz} + I_{xy} I_{zz}^2 + I_{xy} I_{yy} I_{zz} - 2I_{xy} I_{xz}^2) \Delta, \\
 I_{qpr} &= (I_{yz}^2 I_{xx} + I_{zz}^2 I_{xx} + I_{xz}^2 I_{xx} - I_{xx}^2 I_{zz} - I_{xy}^2 I_{zz} - I_{xz}^2 I_{zz}) \Delta,
 \end{aligned}$$

$$\begin{aligned}
 I_{rpp} &= (I_{yy} I_{xy} I_{xx} - I_{xz} I_{yz} I_{xx} - I_{xy} I_{xz}^2 - I_{xy}^3) \Delta, \\
 I_{rqq} &= (I_{xy} I_{yz}^2 + I_{yz} I_{xz} I_{yy} + I_{xy}^3 - I_{xx} I_{xy} I_{yy}) \Delta, \\
 I_{rrr} &= (I_{xx} I_{xz} I_{yz} - I_{xy} I_{yz}^2 - I_{yz} I_{xz} I_{yy} + I_{xy} I_{xz}^2) \Delta, \\
 I_{rpq} &= (I_{yy} I_{xx}^2 + I_{yy} I_{xz}^2 + I_{yy} I_{xy}^2 - I_{yy}^2 I_{xx} - I_{xy}^2 I_{xx} - I_{yz}^2 I_{xx}) \Delta, \\
 I_{rqr} &= (I_{xy} I_{yz} I_{yy} + I_{xy} I_{yz} I_{xx} - I_{xz} I_{yy} I_{xx} + I_{xz} I_{yy}^2 - I_{xz} I_{yy} I_{zz} - I_{yz} I_{xy} I_{zz} + 2I_{xz} I_{xy}^2) \Delta, \\
 I_{rpr} &= (I_{yz} I_{zz} I_{xx} + I_{yz} I_{yy} I_{xx} - I_{xy} I_{xz} I_{xx} - I_{yz} I_{xx}^2 - I_{xz} I_{yy} I_{xy} + I_{xy} I_{zz} I_{xz} - 2I_{xy}^2 I_{yz}) \Delta,
 \end{aligned}$$

et

$$\begin{aligned}
 \dot{I}_{pp} &= (-\dot{I}_{xx}(I_{yy} I_{zz} - I_{yz}^2) + \dot{I}_{xy}(I_{xz} I_{yz} + I_{xy} I_{zz}) + \dot{I}_{xz}(I_{xy} I_{yz} + I_{xz} I_{yy})) \Delta, \\
 \dot{I}_{pq} &= (-\dot{I}_{xy}(I_{yz}^2 - I_{yy} I_{zz}) - \dot{I}_{yy}(I_{xy} I_{zz} + I_{xz} I_{yz}) + \dot{I}_{yz}(I_{xy} I_{yz} + I_{xz} I_{yy})) \Delta, \\
 \dot{I}_{pr} &= (-\dot{I}_{xz}(I_{yz}^2 - I_{yy} I_{zz}) + \dot{I}_{yz}(I_{xy} I_{zz} + I_{xz} I_{yz}) - \dot{I}_{zz}(I_{xy} I_{yz} + I_{xz} I_{yy})) \Delta, \\
 \dot{I}_{qp} &= (-\dot{I}_{xx}(I_{xy} I_{zz} + I_{xz} I_{yz}) - \dot{I}_{xy}(I_{xz}^2 - I_{xx} I_{zz}) + \dot{I}_{xz}(I_{xx} I_{yz} + I_{xy} I_{xz})) \Delta, \\
 \dot{I}_{qq} &= (-\dot{I}_{yy}(I_{xx} I_{zz} - I_{xz}^2) + \dot{I}_{xy}(I_{xy} I_{zz} + I_{xz} I_{yz}) + \dot{I}_{yz}(I_{xx} I_{yz} + I_{xy} I_{xz})) \Delta, \\
 \dot{I}_{qr} &= (-\dot{I}_{yz}(I_{xz}^2 - I_{xx} I_{zz}) + \dot{I}_{xz}(I_{xy} I_{zz} + I_{xz} I_{yz}) - \dot{I}_{zz}(I_{xx} I_{yz} + I_{xy} I_{xz})) \Delta, \\
 \dot{I}_{rp} &= (-\dot{I}_{xx}(I_{xy} I_{yz} + I_{xz} I_{yy}) + \dot{I}_{xy}(I_{xx} I_{yz} + I_{xy} I_{xz}) - \dot{I}_{xz}(I_{xy}^2 - I_{xx} I_{yy})) \Delta, \\
 \dot{I}_{rq} &= (-\dot{I}_{yy}(I_{xx} I_{yz} + I_{xy} I_{xz}) + \dot{I}_{xy}(I_{xy} I_{yz} + I_{xz} I_{yy}) - \dot{I}_{yz}(I_{xy}^2 - I_{xx} I_{yy})) \Delta, \\
 \dot{I}_{rr} &= (-\dot{I}_{zz}(I_{xx} I_{yy} - I_{xy}^2) + \dot{I}_{xz}(I_{xy} I_{yz} + I_{xz} I_{yy}) + \dot{I}_{yz}(I_{xx} I_{yz} + I_{xy} I_{xz})) \Delta,
 \end{aligned}$$

où l'expression de  $\Delta$  est donnée par l'équation (C.30).

### C.2.5 Équations additionnelles

Bien que la dynamique du véhicule soit complètement définie par le jeu de douze équations obtenu dans la section précédente (équations de position, de vitesse, de vitesse angulaire et d'attitude), il est parfois utile de connaître certaines données supplémentaires lors de la synthèse de lois de commande par exemple, comme les angles d'Euler et/ou les composantes de la vitesse dans le trièdre véhicule.

- les angles d'Euler définissent un ensemble ordonné de trois rotations successives permettant de passer du trièdre véhicule au trièdre géocentrique. En utilisant la convention classique pour les véhicules aérospatiaux, ces trois rotations sont définies par 1) une rotation en roulis  $\phi$ , 2) une rotation en assiette longitudinale  $\theta$ , et 3) une rotation en azimut  $\psi$ . Les angles d'Euler sont liés au vecteur  $\vec{\omega} = [p, q, r]^T$  par la relation

$$\begin{bmatrix} \phi \\ \theta \\ \psi \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ \sin \phi \tan \theta & \cos \phi & -\sin \phi \sec \theta \\ \cos \phi \tan \theta & -\sin \phi & -\cos \phi \sec \theta \end{bmatrix} \begin{bmatrix} p \\ q \\ r \end{bmatrix}.$$

- les composantes  $u$ ,  $v$  et  $w$  de la vitesse aérodynamique dans le trièdre véhicule sont simplement données par les expressions

$$\begin{cases} u = V \cos \alpha \cos \beta, \\ v = V \sin \beta, \\ w = V \sin \alpha \cos \beta. \end{cases}$$



# Bibliographie

- [1] M. Fliess, J. Lévine, P. Martin, P. Rouchon, Flatness and defect of non-linear systems : introduction theory and examples, *Int. Journal of Control* 61 (6) (1995) 1327–1361.
- [2] A. Isidori, *Nonlinear Control Systems*, 2nd Edition, Springer Verlag, New York, 1989.
- [3] B. Charlet, J. Lévine, R. Marino, Sufficient conditions for dynamic state feedback linearization, *SIAM J. Control and Optimization* 29 (1991) 38–57.
- [4] M. J. V. Nieuwstadt, R. M. Murray, Real-time trajectory generation for differentially flat systems, *Int. Journal of Robust and Nonlinear Control* 8 (11) (1998) 995–1020.
- [5] M. B. Milam, R. Franz, J. E. Hauser, R. M. Murray, Receding horizon control of a vectored thrust flight experiment, *IEE Proceedings on Control Theory and Applications* 152 (3) (2005) 340–348.
- [6] M. Fliess, C. Join, H. Sira-Ramirez, Non-linear estimation is easy, *Int. J. Modelling, Identification and Control* 3 (2008) 219–227.
- [7] J. Lévine, D. V. Nguyen, Flat output characterization for linear systems using polynomial matrices, *Systems & Control Letters* 48 (1) (2003) 69–75.
- [8] B. Jakubczyk, W. Respondek, On linearization of control systems, *Bull. Acad. Pol. Sci., ser. Sci. Math.* 28 (1980) 517–522.
- [9] L. R. Hunt, R. Su, G. Meyer, Design for multi-input nonlinear systems, in : *Differential Geometric Control Theory*, R. W. Brockett, R. S. Millman, H. J. Sussmann, Birkhauser, Boston, 1983, pp. 268–298.
- [10] A. Glumineau, C. H. Moog, The essential orders and the nonlinear decoupling problem, *Int. J. Control* 50 (6) (1989) 1825–1834.
- [11] P. Mai, C. Join, J. Reger, Flatness-based fault tolerant control of a nonlinear MIMO system using algebraic derivative estimation, in : *Proc. of the 3rd IFAC Symposium on System, Structure and Control*, IFAC, NFOz do Iguaçu, Brazil, 2007.
- [12] N. Faiz, S. K. Agrawal, R. M. Murray, Trajectory planning of differentially flat systems with dynamics and inequalities, *Journal of Guidance, Control and Dynamics* 24 (2) (2001) 219–227.
- [13] J. von Lowis, J. Rudolph, Real-time trajectory generation for flat systems with constraints, Vol. 281 of *Lecture Notes in Control and Information Sciences*, Springer, London, 2003, pp. 385–394.

- [14] M. Fliess, C. Join, H. Mounier, An introduction to nonlinear fault diagnosis with an application to a congested internet router, Lecture Notes in Control and Information Sciences, Springer, London, 2004.
- [15] R. Martinez-Guerra, R. Gonzalez-Galan, A. Luviano-Juarez, J. Cruz-Victoria, Diagnosis for a class of non-differentially flat and liouvillian systems, IMA J. Math. Control & Information 24 (23) (2007) 177–195.
- [16] J. Lévine, On necessary and sufficient conditions for differential flatness, arXiv (0605405). [arXiv:arXiv:math/0605405v2](https://arxiv.org/abs/math/0605405v2).
- [17] P. Martin, Contribution à l'étude des systèmes différentiellement plats, Ph.D. thesis, Ecole Nationale Supérieure des Mines de Paris, Paris, France (1992).
- [18] H. Mounier, Propriétés structurelles des systèmes linéaires à retards : aspects théoriques et pratiques, Ph.D. thesis, Université Paris-sud, Orsay, France (1995).
- [19] N. Petit, Systèmes à retards. platitude en génie des procédés et contrôle de certaines équations des ondes, Ph.D. thesis, Ecole des Mines de Paris, Paris, France (2000).
- [20] J. Rudolph, Flatness Based Control of Distributed Parameter Systems, Shaker Verlag, Aachen, Germany, 2003.
- [21] E. Aranda-Bricaire, C. H. Moog, J. B. Pomet, A linear algebraic framework for dynamic feedback linearization, IEEE Trans. Automat. Contr. 40 (1995) 127–132.
- [22] J. B. Pomet, A differential geometric setting for dynamic equivalence and dynamic linearization, Geometry in Nonlinear Control and Differential Inclusions 32 (1994) 319–339.
- [23] P. S. P. da Silva, Flatness of nonlinear control systems : a Cartan-Kahler approach, in : Proc. Mathematical Theory of Networks and Systems (MTNS'2000), Perpignan, France, 2000, pp. 1–10.
- [24] F. Chyzak, A. Quadrat, D. Robertz, Effective algorithms for parametrizing linear control systems over Ore algebras, Vol. 16, Springer Verlag, Berlin, Heidelberg, 2005, pp. 319–376.
- [25] P. Rocha, J. C. Willems, Behavioral controllability of delay-differential systems, SIAM Journal on Control and Optimization (1997) 254–264.
- [26] J. Lévine, *Analyse et commande des systèmes non linéaires*, Notes de cours, Ecole Nationale Supérieure des Ponts et Chaussées (2004).  
URL <http://cas.ensmp.fr/~levine/index.html>
- [27] H. Mounier, J. Rudolph, M. Fliess, P. Rouchon, Tracking control of a vibrating string with an interior mass viewed as delay system, ESAIM COCV 3 (1998) 315–321.
- [28] M. Fliess, H. Mounier, P. Rouchon, J. Rudolph, Systèmes linéaires sur les opérateurs de mikusiński et commande d'une poutre flexible, in : ESAIM Proc. Conf. Elasticité, viscolélasticité et contrôle optimal, 8ème entretiens du centre Jacques Cartier, Lyon, France, 1996, pp. 157–168.
- [29] A. Quadrat, D. Robertz, Computation of bases of free modules over the weyl algebras, Journal of Symbolic Computation 42 (2007) 1113–1141.

- [30] F. Antritter, J. Lévine, Towards a computer algebraic algorithm for flat output determination, in : International Symposium on Symbolic and Algebraic Computation, Hagenberg, Austria, 2008.
- [31] M. Fliess, J. Lévine, P. Martin, P. Rouchon, A Lie-Bäcklund approach to equivalence and flatness of nonlinear systems, *IEEE Trans. Automatic Control* 44 (5) (1999) 922–937.
- [32] N. X. Vinh, A. Busemann, R. D. Culp, Hypersonic and Planetary Entry Flight Mechanics, The university of Michigan Press, 1980.
- [33] T. Neckel, C. Talbot, N. Petit, Collocation and inversion for a reentry optimal control problem, in : Proc. of the 5th Intern. Conference on Launcher Technology, Madrid, Spain, 2003.
- [34] J. T. Betts, Practical methods for optimal control using nonlinear programming, SIAM, 2001.
- [35] J. D. Schierman, J. R. Hull, D. Ward, On-line trajectory command reshaping for reusable launch vehicles, in : Guidance, Navigation, and Control Conference and Exhibit, AIAA, Austin, Texas, 2003.
- [36] J. R. Hull, N. Gandhi, J. D. Schierman, In-flight taem/final approach trajectory generation for reusable launch vehicles, in : IEEE Aerospace Conference, Big Sky, Montana, 2005.
- [37] P. J. Shaffer, I. M. Ross, M. W. Oppenheimer, D. B. Doman, K. P. Bollino, Fault-tolerant optimal trajectory generation for reusable launch vehicles, *Journal of Guidance, Control and Dynamics* 30 (6) (2007) 1794–1802.
- [38] A. Trent, R. Iyer, D. Doman, Trajectory planning for a re-entry vehicle under failure conditions, in : American Control Conference, IEEE, New York, NY, 2007.
- [39] P. Deuffhard, Newton Methods for Nonlinear Problems. Affine Invariance and Adaptive Algorithms, Vol. 35 of Springer Series in Computational Mathematics, Springer, Berlin, 2004.
- [40] H. W. Kuhn, A. W. Tucker, Nonlinear programming, in : Proceedings of 2nd Berkeley Symposium, Berkeley, California, 1951, pp. 481–492.
- [41] D. G. Luenberger, Linear and Nonlinear Programming, 2nd Edition, Addison-Wesley, Reading, MA, 1989.
- [42] T. Mathworks, Matlab optimization toolbox user’s guide (2009).
- [43] P. E. Gill, W. Murray, M. A. Saunders, SNOPT : An SQP algorithm for large-scale constrained optimization, Tech. rep., Department of Operations Research, Stanford University (1997).
- [44] P. E. Gill, W. Murray, M. A. Saunders, M. H. Wright, User’s guide for NPSOL (version 4.0) : A fortran package for nonlinear programming, Tech. rep., Department of Operations Research, Stanford University (1986).
- [45] K. Holmström, M. M. Edvall, User’s guide for TOMLAB/MINOS, Tech. rep., Tomlab Optimization Inc. (2008).



- [46] R. Fletcher, S. Leyffer, User manual for filterSQP, Tech. rep., University of Dundee (1998).
- [47] R. H. Byrd, J. Nocedal, R. A. Waltz, KNITRO : An Integrated Package for Nonlinear Optimization, Large-Scale Nonlinear Optimization, Springer-Verlag, New York, 2006, pp. 35–59.
- [48] L. S. Pontryagin, V. G. Boltyanskii, R. V. Gamkrelidze, E. F. Mishchenko, The Mathematical Theory of Optimal Processes, Vol. 4 of L. S. Pontryagin Selected Works, Gordon and Breach, Montreux, Switzerland, 1986.
- [49] U. Ascher, R. Mattheij, Numerical solution of boundary value problems in ordinary differential equations, Prentice-Hall, Englewoods Cliffs, NJ, 1988.
- [50] P. Lu, Predictor-corrector entry guidance for low-lifting vehicles, *Journal of Guidance, Control and Dynamics* 31 (4) (2008) 1067–1075.
- [51] A. Saraf, J. A. Leavitt, D. T. Chen, K. D. Mease, Design and evaluation of an acceleration guidance algorithm for entry, *Journal of Guidance, Control and Dynamics* 41 (6) (2004) 986–996.
- [52] J. A. Leavitt, A. Saraf, D. T. Chen, K. D. Mease, Performance of evolved acceleration guidance logic for entry (eagle), in : *Guidance, Navigation, and Control Conference and Exhibit*, AIAA, Monterey, California, 2002.
- [53] K. D. Mease, D. T. Chen, A three-dimensional predictive entry guidance approach, in : *Guidance, Navigation, and Control Conference and Exhibit*, AIAA, Denver, Colorado, 2000.
- [54] G. H. Barton, S. G. Tragesser, Auto-landing trajectory design for the X-34, in : *Atmospheric Flight Mechanics Conference and Exhibit*, AIAA, Portland, Oregon, 1999, pp. 15–24.
- [55] R. Bulirsch, Die mehrzielmethode zur numerischen lösung von nichtlinearen randwertproblemen und aufgaben der optimalen steuerung, Tech. rep., Carl-Cranz-Gesellschaft, Oberpfaffenhofen, Germany (1971).
- [56] H. Kreim, B. Kugelmann, H. J. Pesch, M. H. Breitner, Minimizing the maximum heating of a reentering space shuttle : An optimal control problem with multiple control constraints, *Optimal Control Applications and Methods* 17 (1996) 45–69.
- [57] Z. Shen, P. Lu, Onboard generation of three-dimensional constrained entry trajectories, *Journal of Guidance, Control and Dynamics* 26 (1) (2003) 111–121.
- [58] C. A. Kluever, Unpowered approach and landing guidance using trajectory planning, *Journal of Guidance, Control and Dynamics* 27 (6) (2004) 967–974.
- [59] K. R. Horneman, C. A. Kluever, Terminal area energy management trajectory planning for an unpowered reusable launch vehicle, in : *Atmospheric Flight Mechanics Conference and Exhibit*, AIAA, Providence, Rhode Island, 2004.
- [60] C. A. Kluever, K. R. Horneman, Terminal trajectory planning and optimization for an unpowered reusable launch vehicle, in : *Guidance, Navigation, and Control Conference and Exhibit*, AIAA, San Francisco, California, 2005.

- [61] A. J. Calise, N. Brandt, Generation of launch vehicle abort trajectories using a hybrid optimization method, *Journal of Guidance, Control and Dynamics* 27 (6) (2004) 930–937.
- [62] P. Lu, Entry guidance and trajectory control for reusable launch vehicle, *Journal of Guidance, Control and Dynamics* 20 (1) (1997) 143–149.
- [63] J. D. Schierman, J. R. Hull, In-flight entry trajectory optimization for reusable launch vehicles, in : *Guidance, Navigation, and Control Conference and Exhibit*, AIAA, San Francisco, California, 2005.
- [64] I. M. Ross, F. Fahroo, Issues in the real-time computation of optimal control, *Mathematical and computer modelling* 43 (9-10) (2006) 1172–1188.
- [65] K. Hornik, M. Stinchcombe, H. White, Multilayer feedforward networks are universal approximators, *Neural Networks* 2 (5) (1989) 359–366.
- [66] G. Dreyfus, M. Samuelides, J. M. Martinez, M. Gordon, F. Badran, S. Thiria, L. Héroult, *Réseaux de neurones*, Eyrolles, 2004.
- [67] B. T. Burchett, Fuzzy logic trajectory design and guidance for terminal area energy management, *AIAA Journal of Guidance, Control and Dynamics* (700).
- [68] G. Winter, J. Periaux, M. Galan, P. Cuesta, *Genetic Algorithms in Engineering and Computer Science*, John Wiley & Sons, New York, 1996.
- [69] N. Yokoyama, S. Suzuki, Modified genetic algorithm for constrained trajectory optimization, *Journal of Guidance, Control and Dynamics* 28 (1) (2005) 139–144.
- [70] L. A. Zadeh, Fuzzy sets, *Information and Control* 8 (3) (1965) 338–353.
- [71] K. M. Passino, S. Yurkovich, *Fuzzy Control*, Addison-Wesley, 1998.
- [72] J. D. Schierman, D. G. Ward, J. F. Monaco, J. R. Hull, A reconfigurable guidance approach for reusable launch vehicles, in : *Guidance, Navigation, and Control Conference and Exhibit*, AIAA, Montreal, Canada, 2001.
- [73] Z. Shen, P. Lu, Dynamic lateral entry guidance logic, *Journal of Guidance, Control and Dynamics* 27 (6) (2004) 949–959.
- [74] C. Brunner, P. Lu, Skip entry trajectory planning and guidance, *Journal of Guidance, Control and Dynamics* 31 (5) (2008) 1210–1219.
- [75] P. Lu, D. B. Doman, J. D. Schierman, Adaptive terminal guidance for hypervelocity impact in specified direction, *Journal of Guidance, Control and Dynamics* 29 (2) (2006) 269–278.
- [76] A. Saraf, J. A. Leavitt, K. D. Mease, Landing footprint computation for entry vehicles, in : *Guidance, Navigation, and Control Conference and Exhibit*, AIAA, Providence, Rhode Island, 2004.
- [77] K. D. Mease, D. T. Chenz, P. Teufel, H. Schonenberger, Reduced-order entry trajectory planning for acceleration guidance, *Journal of Guidance, Control and Dynamics* 25 (2) (2002) 257–266.
- [78] J. A. Leavitt, K. D. Mease, Feasible trajectory generation for atmospheric entry guidance, *Journal of Guidance, Control and Dynamics* 30 (2) (2007) 473–481.

- [79] E. N. Johnson, A. J. Calise, M. D. Curry, K. D. Mease, J. E. Corban, Adaptive guidance and control for autonomous hypersonic vehicles, *Journal of Guidance, Control and Dynamics* 29 (3) (2006) 725–737.
- [80] T. Buchner, Trajectory generation strategy for the terminal area of a reusable launch vehicle, in : 54th International Astronautical Congress of the International Astronautical Federation, the International Academy of Astronautics, and the International Institute of Space Law, IAC, Bremen, Germany, 2003.
- [81] W. Grimm, J. G. van der Meulen, A. J. Roenneke, Optimal update scheme for drag reference profiles in an entry guidance, *Journal of Guidance, Control and Dynamics* 26 (5) (2003) 695–701.
- [82] A. Joshi, K. Sivan, S. S. Amma, Predictor-corrector reentry guidance algorithm with path constraints for atmospheric entry vehicles, *Journal of Guidance, Control and Dynamics* 30 (5) (2007) 1307–1318.
- [83] A. D. Ngo, D. B. Doman, Footprint determination for reusable launch vehicles experiencing control effector failures, in : *Guidance, Navigation, and Control Conference and Exhibit*, AIAA, Monterey, California, 2002.
- [84] R. Holsapple, R. Venkataraman, D. B. Doman, New, fast numerical method for solving two-point boundary-value problems, *Journal of Guidance, Control and Dynamics* 27 (2) (2004) 301–304.
- [85] T. E. Moore, Space shuttle entry terminal area energy management, Tech. Rep. 104744, NASA Technical Memorandum (November 1991).
- [86] A. Trent, R. Venkataraman, D. Doman, Trajectory generation using a modified simple shooting method, in : *IEEE Aerospace Conference, Big Sky, Montana*, 2004.
- [87] C. A. Kluever, Terminal guidance for an unpowered reusable launch vehicle with bank constraints, *Journal of Guidance, Control and Dynamics* 30 (1) (2007) 162–168.
- [88] C. A. Kluever, Unpowered approach and landing guidance with normal acceleration limits, *Journal of Guidance, Control and Dynamics* 30 (3) (2007) 882–885.
- [89] C. A. Kluever, Entry guidance using analytical atmospheric skip trajectories, *Journal of Guidance, Control and Dynamics* 31 (5) (2008) 1531–1534.
- [90] Z. Shen, P. Lu, On-board entry trajectory planning expanded to sub-orbital flight, in : *Guidance, Navigation, and Control Conference and Exhibit*, AIAA, Austin, Texas, 2003.
- [91] A. Joshi, K. Sivan, B. N. Suresh, S. S. Amma, A generic reentry trajectory planning algorithm for reusable launch vehicle missions, in : *NSC conference*, Bombay, India, 2005.
- [92] J. T. A. Chartres, M. H. Grasslin, G. Schneider, Optimization of the terminal flight phase for a future reusable launch vehicle, in : *Guidance, Navigation, and Control Conference and Exhibit*, AIAA, San Francisco, California, 2005.
- [93] P. Vernis, E. Ferreira, On-board trajectory planner for the TAEM guidance of a winged-body, in : *4th AAAF International Symposium on Atmospheric Reentry, Vehicles and Systems*, AAAF, Arcachon, France, 2005.

- [94] R. R. da Costa, Studies for terminal area GNC of reusable launch vehicles, *AIAA Journal of Guidance, Control and Dynamics* (5438).
- [95] F. M. C. Camara, Development and design of the terminal area energy management guidance for a reusable launch vehicle, in : 13th International Space Planes and Hypersonics Systems and Technologies, AIAA/CIRA, Capua, Italia, 2005.
- [96] R. F. Stengel, Optimal guidance for the space shuttle transition, *Journal of Spacecraft and Rockets* 11 (3) (1974) 173–179.
- [97] D. A. Allwine, J. E. Fisher, J. A. Strahler, D. A. Lawrence, M. W. Oppenheimer, D. B. Doman, On-line trajectory generation for hypersonic vehicles, in : Guidance, Navigation, and Control Conference and Exhibit, AIAA, San Francisco, California, 2005.
- [98] P. Lu, Regulation about time-varying trajectories : Precision entry guidance illustrated, *Journal of Guidance, Control and Dynamics* 22 (6) (1999) 784–790.
- [99] K. D. Mease, P. Teufel, H. Schonenberger, D. T. Chen, S. Bharadwajz, Re-entry trajectory planning for a reusable launch vehicle, in : Atmospheric and Flight Mechanics Conference, AIAA, Portland, Oregon, 1999.
- [100] K. P. Bollino, I. M. Ross, D. B. Doman, Optimal nonlinear feedback guidance for reentry vehicles, in : Guidance, Navigation, and Control Conference and Exhibit, AIAA, Keystone, Colorado, 2006.
- [101] K. Graichen, N. Petit, Constructive methods for initialization and handling mixed state-input constraints in optimal control, *Journal of Guidance, Control and Dynamics* 31 (5) (2008) 1334–1343.
- [102] K. Graichen, N. Petit, A continuation approach to state and adjoint calculation in optimal control applied to the reentry problem, in : 7th IFAC World Congress, Seoul, Korea, 2008, pp. 14307–14312.
- [103] J. F. Bonnans, G. Launay, Large-scale direct optimal control applied to a reentry problem, *Journal of Guidance, Control and Dynamics* 21 (6) (1998) 996–1000.
- [104] P. N. Desai, B. A. Conway, Six-degree-of-freedom trajectory optimization using a two-timescale collocation architecture, *Journal of Guidance, Control and Dynamics* 31 (5) (2008) 1308–1315.
- [105] S. Jain, P. Tsiotras, Trajectory optimization using multiresolution techniques, *Journal of Guidance, Control and Dynamics* 31 (5) (2008) 1424–1436.
- [106] J. M. C. III, M. S. Epstein, D. G. MacMynowski, R. M. Murray, Optimal nonlinear guidance with inner-loop feedback for hypersonic re-entry, in : American Control Conference, IFAC, Minneapolis, Minnesota, 2006.
- [107] J. D. Schierman, D. G. Ward, J. R. Hull, J. F. Monaco, M. J. Ruth, Adaptive guidance systems for hypersonic reusable launch vehicles, in : IEEE Aerospace Conference, AIAA, Big Sky, Montana, 2001.
- [108] J. D. Schierman, J. R. Hull, D. G. Ward, Adaptive guidance with trajectory reshaping for reusable launch vehicles, in : Guidance, Navigation, and Control Conference and Exhibit, AIAA, Monterey, California, 2002.

- [109] J. D. Schierman, D. G. Ward, J. R. Hull, N. Gandhi, Intelligent guidance and trajectory command systems for autonomous space vehicles, in : 1st Intelligent Systems Technical Conference, AIAA, Chicago, Illinois, 2004.
- [110] K. Grantham, Adaptive critic neural network based terminal area energy management/entry guidance, AIAA Journal of Guidance, Control and Dynamics (305).
- [111] K. Grantham, Adaptive critic neural network based terminal area energy management/entry guidance, in : 41st Aerospace Sciences Meeting and Exhibit, Reno, Nevada, 2003.
- [112] R. K. Arora, Reentry trajectory optimization : Evolutionary approach, in : 9th AIAA/ISSMO Symposium on Multidisciplinary Analysis and Optimization, Atlanta, Georgia, 2002.
- [113] J. Luo, W. Zhou, Q. Fang, J. Yuan, An integrated optimization of RLV reentry trajectory, in : Intelligent Automation Conference, IEEE, Hong-Kong, 2003.
- [114] C. Gang, X. Min, W. Zi-Ming, C. Si-lu, RLV reentry trajectory multi-objective optimization design based on NSGA-II algorithm, in : Atmospheric Flight Mechanics Conference and Exhibit, AIAA, San Francisco, California, 2005.
- [115] C. Gang, W. Zi-Ming, X. Min, C. Si-lu, Genetic algorithm optimization of RLV reentry trajectory, in : 13th International Space Planes and Hypersonics Systems and Technologies, AIAA/CIRA, Capua, Italia, 2005.
- [116] B. T. Burchett, Fuzzy logic trajectory design and guidance for terminal area energy management, Journal of spacecraft and rockets 41 (3) (2004) 444–450.
- [117] B. T. Burchett, Genetic algorithm tuned fuzzy logic for gliding return trajectories, in : 42nd Aerospace Sciences Meeting and Exhibit, AIAA, Reno, Nevada, 2004.
- [118] M. Bikdash, K. Sartor, A. Homaifar, Fuzzy guidance of the shuttle orbiter during atmospheric reentry, Control Engineering Practice 7 (1999) 295–303.
- [119] C. A. Schmitt, B. T. Burchett, Fuzzy IPPD guidance for approach and landing of a reusable launch vehicle, in : 1st Intelligent Systems Technical Conference, AIAA, Chicago, Illinois, 2004.
- [120] J. Laurent-Varin, J. F. Bonnans, N. Bérend, M. Haddou, C. Talbot, Interior-point approach to trajectory optimization, Journal of Guidance, Control and Dynamics 30 (5) (2007) 1228–1238.
- [121] P. Apkarian, P. Gahinet, A convex characterization of gain-scheduled  $H_\infty$  controllers, IEEE Trans. Automatic Control 40 (5) (1995) 853–864.
- [122] M. Zerar, F. Cazaurang, A. Zolghadri, Coupled lpv and flatness-based approach for space reentry vehicles guidance, IET Control Theory and Applications.
- [123] A. Marcos, G. J. Balas, Development of linear-parameter-varying models for aircraft, Journal of Guidance, Control and Dynamics 27 (2) (2004) 218–228.
- [124] V. Morio, F. Cazaurang, P. Vernis, Flatness-based hypersonic reentry guidance of a lifting-body vehicle, Control Engineering Practice 17 (5) (2009) 588–596.
- [125] V. Morio, P. Vernis, F. Cazaurang, Hypersonic reentry and flatness theory - application to medium L/D entry vehicle, in : Proc. of the 1st International ARA Days, Arcachon, France, 2006.

- [126] V. Morio, F. Cazaurang, A. Falcoz, P. Vernis, Robust terminal area energy management guidance using flatness approach, *IET Control Theory & Applications*.
- [127] V. Morio, A. Falcoz, P. Vernis, F. Cazaurang, On the design of a flatness-based guidance algorithm for the terminal area energy management of a winged-body vehicle, in : *Proc. of the 17th IFAC Symposium on Automatic Control in Aerospace*, IFAC, Toulouse, France, 2007.
- [128] D. Ito, J. Georgie, J. Valasek, D. T. Ward, Reentry vehicle flight controls design guidelines : Dynamic inversion, Tech. Rep. 210771, NASA Technical Report (2002).
- [129] J. Georgie, J. Valasek, Evaluation of longitudinal desired dynamics for dynamic-inversion controlled generic reentry vehicles, *AIAA Journal of Guidance, Control and Dynamics* 26 (5) (2003) 811–819.
- [130] T. R. Bevacqua, Advanced flight control issues for reusable launch vehicles, Ph.D. thesis, Ohio University, Ohio, USA (2004).
- [131] M. B. Milam, K. Mushambi, R. M. Murray, A new computational approach to real-time trajectory generation for constrained mechanical systems, in : *Proc. of the 39th IEEE Conference on Decision and Control*, Vol. 1, Sydney, Australia, 2000, pp. 845–851.
- [132] M. B. Milam, N. Petit, R. M. Murray, Constrained trajectory generation for micro-satellite formation flying, in : *AIAA Guidance, Navigation and Control Conference*, 2001, pp. 328–333.
- [133] M. B. Milam, Real-time optimal trajectory generation for constrained dynamical systems, Ph.D. thesis, California Institute of Technology, Pasadena, California (2003).
- [134] M. B. Milam, N. Petit, Constrained trajectory generation for a planar missile, Tech. rep., California Institute of Technology, Pasadena, California (2001).
- [135] W. B. Dunbar, M. B. Milam, R. Franz, R. M. Murray, Model predictive control of a thrust-vectoring flight control experiment, in : *Proc. of 15th IFAC World Congress*, Barcelona, Spain, 2002.
- [136] C. Louembet, F. Cazaurang, A. Zolghadri, C. Pittet, C. Charbonnel, Design of algorithms for satellite slew maneuvers by flatness and collocation, in : *Proc. of the American Control Conference*, IEEE, New York City, USA, 2007.
- [137] M. E. Dyer, A. M. Frieze, On the complexity of computing the volume of a polyhedron, *SIAM J. Comput.* 17 (5) (1988) 967–974.
- [138] A. H. Barr, Superquadrics and angle-preserving transformations, *IEEE Computer Graphics and Applications* 1 (1) (1981) 11–23.
- [139] A. J. Hanson, Hyperquadrics : smoothy deformable shapes with convex polyhedral bounds, *Computer vision, graphics, and image processing* 44 (2) (1988) 191–210.
- [140] J. Gielis, A generic geometric transformation that unifies a wide range of natural and abstract shapes, *American Journal of Botany* 90 (3) (2003) 333–338.
- [141] C. Blanc, C. Schlick, Ratioquadrics : an alternative model for superquadrics, *The Visual Computer* 12 (8) (1996) 420–428.



- [142] T. O. Binford, Visual perception by computer, in : Proc. IEEE Conf. Systems and Controls, IEEE, Miami, 1971.
- [143] I. Biederman, Recognition-by-components : A theory of human image understanding, *Psychological Review* 94 (2) (1987) 115–147.
- [144] M. Pilu, R. B. Fisher, Equal-distance sampling of superellipse models, in : Proc. 6th British Machine Vision Conference (BMVC), Birmingham, UK, 1995.
- [145] H. Löffelmann, E. Gröller, Parameterizing superquadrics, in : Third International Conference in Central Europe in Computer Graphics and Visualisation, Plzen, Czech Republic, 1995, pp. 162–172.
- [146] D. Mortari, On the rigid rotation concept in n-dimensional spaces, *The Journal of the astronautical sciences* 49 (3) (2001) 401–420.
- [147] L. G. Roberts, Homogeneous matrix representation and manipulation of N-dimensional constructs. *The Computer Display Review*, C. W. Adams Associates, Inc., Cambridge, Mass, 1965.
- [148] F. Solina, R. Bajcsy, Recovery of parametric models from range images : the case for superquadrics with global deformations, *IEEE Pattern Analysis and Machine Intelligence* 12 (2) (1990) 131–147.
- [149] W. Kaplan, *Advanced Calculus*, Addison-Wesley, 1993, pp. 286–291.
- [150] R. Cerf, Asymptotic convergence of genetic algorithms, *Adv. in Appl. Probab.* 30 (2) (1998) 521–550.
- [151] J. He, X. Yu, Conditions for the convergence of evolutionary algorithms, *Journal of systems architecture* 47 (7) (2001) 601–612.
- [152] C. Darwin, *The Origin of Species*, Gramercy, 1995.
- [153] D. E. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison-Wesley Professional, 1989.
- [154] T. Back, *Evolutionary algorithms in theory and practice : evolution strategies, evolutionary programming, genetic algorithms*, Oxford University Press, Oxford, UK, 1996.
- [155] C. A. C. Coello, A survey of constraint handling techniques used with evolutionary algorithms, Tech. rep., Laboratorio Nacional de Informatica Avanzada (1999).
- [156] K. Deb, An efficient constraint handling method for genetic algorithms, *Computer Methods in Applied Mechanics and Engineering* 186 (2-4) (2000) 311–338.
- [157] K. Deb, R. B. Agrawal, Simulated binary crossover for continuous search space, *Complex Systems* 9 (1995) 115–148.
- [158] K. Deb, M. Goyal, A combined genetic adaptive search (GeneAS) for engineering design, *Computer Science and Informatics* 26 (1996) 30–45.
- [159] H. G. Eggleston, *Convexity*, Cambridge University Press, 1958.
- [160] Q. Gong, F. Fahroo, I. M. Ross, Spectral algorithm for pseudospectral methods in optimal control, *Journal of Guidance, Control and Dynamics* 31 (3) (2008) 460–471.

- [161] M. W. Oppenheimer, D. B. Doman, M. A. Bolender, A method for estimating control failure effects for aerodynamic vehicle trajectory retargeting, in : Guidance, Navigation, and Control Conference and Exhibit, AIAA, Providence, Rhode Island, 2004.
- [162] M. Bodson, Evaluation of optimization methods for control allocation, *Journal of Guidance, Control, and Dynamics* 25 (4) (2002) 703–711.
- [163] O. Härkegåard, Dynamic control allocation using constrained quadratic programming, *Journal of Guidance, Control, and Dynamics* 27 (6) (2004) 1028–1034.
- [164] M. Bolender, D. Doman, Nonlinear control allocation using piecewise linear functions, *Journal of Guidance, Control, and Dynamics* 27 (6) (2004) 1017–1027.
- [165] J. Petersen, M. Bodson, Interior-point algorithms for control allocation, *Journal of Guidance, Control, and Dynamics* 28 (3) (2005) 471–480.
- [166] T. E. Surber, W. E. Bornemann, A. M. Lee, STS-1 aerodynamic design data book, Tech. rep., NASA (1980).
- [167] W. C. Durham, Constrained control allocation - three-moment problem, *Journal of Guidance, Control, and Dynamics* 17 (2) (1994) 330–336.
- [168] M. A. Bolender, D. B. Doman, Method for determination of nonlinear attainable moment sets, *Journal of Guidance, Control, and Dynamics* 27 (5) (2004) 907–913.
- [169] W. C. Durham, Computationally efficient control allocation, *Journal of Guidance, Control, and Dynamics* 24 (3) (2001) 519–524.
- [170] A. Falcoz, D. Henry, A. Zolghadri, E. Bornschleg, M. Ganet, On-board model-based robust FDIR strategy for reusable launch vehicles, in : 7th International ESA Conference on Guidance, Navigation and Control Systems, Tralee, County Kerry, Ireland, 2008.
- [171] A. Falcoz, D. Henry, A. Zolghadri, A nonlinear fault identification scheme for reusable launchvehicles control surfaces, in : 7th IFAC Symposium on Fault Detection, Supervision and Safety of Technical Processes, Barcelona, Spain, 2009.
- [172] C. de Boor, A practical guide to spline, *Applied Mathematical Sciences*, Springer, New York, 2001.
- [173] A. Bushgens, V. Yaroshevsky, A. Vankov, A. Shourov, Lectures about buran and the recent MAKS re-entry, in : ESTEC Workshop on Russian Guidance, Navigation and Control Reentry Technologies, ESA, 1999.
- [174] A. E. Long, C. A. Long, Surface approximation and interpolation via matrix SVD, *The College Mathematics Journal* 32 (1) (2001) 20–25.
- [175] K. Levenberg, A method for the solution of certain problems in least squares, *Quart. Appl. Math* 2 (1944) 164–168.
- [176] D. W. Marquardt, An algorithm for least-squares estimation of nonlinear parameters, *SIAM J. Appl. Math* 11 (1963) 431–441.
- [177] J. T. Kindler, M. Ulrich, K. H. Well, Entry interface window of landing site cooper pedy for the experimental vehicle X-38 V201, in : Atmospheric Flight Mechanics Conference, AIAA, Denver, Colorado, 2000.



- [178] P. Gath, K. H. Well, Automatic initial guess generation for ariane 5 dual payload ascent trajectory optimization, in : Guidance, Navigation, and Control Conference, AIAA, Denver, Colorado, 2000.
- [179] G. Ortega, T. Sgobba, A. Bavandi, S. Weikert, F. Cremaschi, Automatic transfer vehicle ATV reentry safety trajectory analysis with ASTOS, in : Proc. 3rd IAASS Conference, Building A Safer Space Together, Rome, Italy, 2008.
- [180] S. Erb, D. Barbagallo, F. Cremaschi, G. Ortega, A. Martinez, Safety analysis for stage reentry of VEGA LV, in : Proc. 3rd IAASS Conference, Building A Safer Space Together, Rome, Italy, 2008.
- [181] Z. Jiang, R. Ordonez, Trajectory generation on approach and landing for rlvs using motion primitives and neighboring optimal control, in : American Control Conference, New York City, USA, 2007.
- [182] Z. Jiang, R. Ordonez, Robust approach and landing trajectory generation for reusable launch vehicles in winds, in : 17th IEEE International Conference on Control Applications, San Antonio, Texas, 2008, pp. 930–935.
- [183] A. Miele, T. Wang, W. W. Melvin, Penetration landing guidance trajectories in the presence of windshear, *Journal of Guidance, Control, and Dynamics* 12 (6) (1989) 806–814.
- [184] B. Etkin, *Dynamics of Atmospheric Flight*, John Wiley and Sons, New York, 1972.
- [185] K. J. C. Smith, Majority decodable codes derived from finite geometries, Ph.D. thesis, Mimeograph Series 561, Institute of Statistics, Chapel Hill, NC (1967).
- [186] J. C. McConnell, J. C. Robson, *Noncommutative Noetherian Rings*, American Mathematical Society, 2000.
- [187] P. M. Cohn, *Free rings and their relations*, Academic Press, London, 1985.
- [188] M. G. Cox, The numerical evaluation of B-splines, *J. Inst. Mathematics and Applications* 10 (1972) 134–149.
- [189] L. Piegl, W. Tiller, *The Nurbs Book*, Springer, New York, 1997.
- [190] J. L. Boiffier, *The Dynamics of Flight : Equations*, John Wiley & Sons Inc., 1998.
- [191] F. J. Regan, S. M. Anandakrishnan, *Dynamics of Atmospheric Re-Entry*, AIAA Education Series, AIAA, 1993.

# Index

- équations d'Euler-Lagrange, 102
- équilibrage statique, 193
- équivalence de Lie-Bäcklund, 49
- état adjoint, 102
  
- algorithme génétique, 136, 181, 268
- allocation, 192, 288
- analyse en composantes principales, 229
- annexion superellipsoïdale, 180, 267
- anomalie, 157
- aptitude, 181
- arc singulier, 104
- architecture GNC, 143
- atterrissage automatique, 70, 220, 225, 250
  
- B-spline, 206, 301
  
- carte d'équilibrage statique, 194, 282
- champ de cisaillement, 258
- cisaillement de vent, 258
- coefficients aérodynamiques, 229
- collocation, 108, 127, 130
- conditions de saut, 105
- contrainte de résidu, 122, 128, 130
- convexité, 159, 187
- corde vibrante, 47
- croisement binaire simulé, 185
- cylindre d'alignement, 220
  
- décomposition de Smith, 297
- décomposition en valeurs singulières, 229
- déficience d'équilibrage, 194, 283
- dérivée directionnelle, 90
- demi-grand axe, 157
- direction de pincement, 167
- direction de recherche, 89, 93, 98
- distance radiale euclidienne, 178, 180
- données d'apprentissage, 232
  
- ensemble actif, 93
  
- facteur de charge, 222
- fonction d'appartenance, 160, 170
- fonction de mérite, 94
- fonction de pénalité, 95
- fonctionnelle de coût, 150
- forme de Smith, 298
- forte fermeture, 51
  
- guidage autonome, 143
  
- hyperquadrique, 155
  
- inversion dynamique, 145
- isomorphisme de Lie-Bäcklund, 49
  
- jets d'ordre infini, 49, 153
  
- lagrangien, 92, 96
- loi de pilotage, 145
- loi de quadrature, 209
  
- mécanisme d'encodage, 181
- méthode d'Euler, 107, 131
- méthode de Hermite-Simpson, 107, 128, 132
- méthode de Newton, 88, 93, 94
- méthode de Runge-Kutta, 107, 128, 131
- méthode directe, 109
- méthode indirecte, 109
- méthode trapézoïdale, 107, 128, 131
- matrice élémentaire, 298
- matrice unimodulaire, 297
- maximum de Pontryagin, 103, 116
- migration, 186
- Monte Carlo, 136
- moteur d'inférence, 137
- multiplicateur de Lagrange, 92, 93, 99
- mutation paramétrique, 185

- navette spatiale américaine, 215
- navigation, 146
- noeuds de segmentation, 301
  
- observable autonome, 75
- opérateur génétique, 182
  
- paramétrisation angle-centre, 162, 169
- paramétrisation trigonométrique, 158, 168
- paramètre de forme, 157
- paramètre de pénalité, 96
- pas d'intégration, 106
- perturbation aérologique, 256
- phase d'apprentissage, 232
- phase de dissipation d'énergie, 70, 76, 220
- phase de validation, 232
- phase hypersonique, 75, 220
- pincement linéaire, 167
- planificateur de trajectoires, 143
- platitude, 42, 150
- platitude locale, 50
- point de collocation, 129, 207
- point de contrôle, 206, 302
- population initiale, 181
- poursuite de trajectoire, 143
- pression dynamique, 222, 267
- problème aux deux bouts, 103
- problème aux valeurs initiales, 105, 110
- problème de commande optimale, 101, 149, 186, 271
- produit sphérique, 157
- programmation non linéaire, 97, 146
- programmation quad. séquentielle, 88, 98
- programmation quadratique, 94, 98
  
- quadrillage paramétrique, 161
  
- régions de confiance, 96
- réseau de neurones, 135, 229
- recherche linéaire, 96
- rentrée atmosphérique, 70, 75, 214
- replanification de trajectoire, 189, 283
- rotation  $n$ -D, 166
  
- sélection par tournois, 183
- séparation des variables, 229
  
- séquence de segmentation, 301
- séquence nodale, 206, 301
- sortie plate, 43, 51, 151
- superellipse, 156
- superellipsoïde, 156, 267
- superquadrique, 155, 266
- système de Karush-Kuhn-Tucker, 93, 95
- système flou, 137
- système implicite, 49
- système masses-ressorts, 44
- système plat, 50, 151
  
- tableau de Butcher, 106
- technique de tir, 109, 110, 116
- technique de transcription, 109, 127, 130
- Terre plate, 70
- Terre tournante, 76
- tir multiple, 128
- tirs multiples, 121
- translation  $n$ -D, 166
  
- véhicule de rentrée, 214
- véhicule non holonome, 43
- variable de relaxation, 99
- volume, 171, 178

## Résumé

Cette thèse porte sur le développement d'une loi de guidage autonome par platitude pour les véhicules de rentrée atmosphérique. La loi de guidage autonome proposée dans ce mémoire s'appuie sur le concept de platitude, afin d'effectuer un traitement des informations à bord, dans le but double d'attribuer un niveau de responsabilité et d'autonomie au véhicule, déchargeant ainsi le segment sol de tâches opérationnelles "bas niveau", pour lui permettre de mieux assumer son rôle de coordination globale. La première partie de ce mémoire traite de la caractérisation formelle de sorties plates pour les systèmes non linéaires régis par des équations différentielles ordinaires, ainsi que pour les systèmes linéaires à retards. Un algorithme constructif est proposé afin de calculer des sorties  $\delta$ -plates candidates sous un environnement de calcul formel standard. Dans la seconde partie, une méthodologie complète et générique de replanification de trajectoires de rentrée atmosphérique est proposée, afin de doter le véhicule d'un certain niveau de tolérance à des défauts d'actionneurs simple/multiples, pouvant survenir lors des phases critiques d'une mission de rentrée atmosphérique. En outre, une méthodologie d'annexion superellipsoïdale est proposée afin de convexifier le problème de commande optimale décrit dans l'espace des sorties plates. La loi de guidage autonome proposée est ensuite appliquée étape par étape à une mission de rentrée atmosphérique pour la navette spatiale américaine STS-1.

**Mots-clés:** rentrée atmosphérique, platitude, guidage autonome, replanification de trajectoires.

## Abstract

This thesis deals with the design of an autonomous guidance law based on flatness approach for atmospheric reentry vehicles. The autonomous guidance law proposed in this dissertation is based on flatness concept, in order to perform onboard processing so as to locally assign autonomy and responsibility to the vehicle, thus exempting the ground segment from "low level" operational tasks, so that it can ensure more efficiently its mission of global coordination. The first part of the manuscript deals with the formal characterization of flat outputs for nonlinear systems governed by ordinary differential equations, as well as for linear time-delay systems. A constructive algorithm is proposed in order to compute candidate  $\delta$ -flat outputs within a standard formal computing environment. In the second part of the manuscript, a global and generic reentry trajectory replanning methodology is proposed in order to provide a fault-tolerance capability to the vehicle when subject to single/multiple control surface failures that could occur during the critical phases of an atmospheric reentry mission. In addition, a superellipsoidal annexion method is proposed so as to convexify the optimal control problem described in the flat outputs space. The proposed autonomous guidance law is then applied step by step to an atmospheric reentry mission for the US Space Shuttle orbiter STS-1.

**Keywords:** atmospheric reentry, flatness, autonomous guidance, trajectory replanning.

