



**HAL**  
open science

# Méthode à deux niveaux et préconditionnement géométrique en contrôle optimal. Application au problème de répartition de couple des véhicules hybrides électriques

Rémy Dutto

## ► To cite this version:

Rémy Dutto. Méthode à deux niveaux et préconditionnement géométrique en contrôle optimal. Application au problème de répartition de couple des véhicules hybrides électriques. Mathématiques [math]. Université de Toulouse, 2024. Français. NNT : 2024TLSEP088 . tel-04792906

**HAL Id: tel-04792906**

**<https://theses.hal.science/tel-04792906v1>**

Submitted on 20 Nov 2024

**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.

# Doctorat de l'Université de Toulouse

préparé à Toulouse INP

---

Méthode à deux niveaux et préconditionnement géométrique en contrôle optimal. Application au problème de répartition de couple des véhicules hybrides électriques.

---

Thèse présentée et soutenue, le 7 octobre 2024 par

**Rémy DUTTO**

## École doctorale

EDMITT - Ecole Doctorale Mathématiques, Informatique et Télécommunications de Toulouse

## Spécialité

Mathématiques et Applications

## Unité de recherche

IRIT : Institut de Recherche en Informatique de Toulouse

## Thèse dirigée par

Olivier COTS et Mariano SANS

## Composition du jury

M. Jean-Baptiste CAILLAU, Président, Université Côte d'Azur

M. Alain RAPAPORT, Rapporteur, INRAE Occitanie-Montpellier

Mme Hasnaa ZIDANI, Rapporteur, INSA Rouen

M. Michel POVLOVITSCH SEIXAS, Examineur, Continental

M. Olivier COTS, Directeur de thèse, Toulouse INP

M. Mariano SANS, Co-directeur de thèse du monde socio-économique, Vitesco Technologies



*À ma tatie*

« Je ne cherche pas à connaître les réponses,  
je cherche à comprendre les questions. »  
Confucius



# Table des matières

---

<b>Remerciements</b>	<b>ix</b>
<b>Introduction</b>	<b>xi</b>
<b>Chapitre 1. Application</b>	<b>1</b>
1.1 Description du problème	2
1.1.1 Cycle	2
1.1.2 Relation entrée-sortie	5
1.1.3 Variables importantes	6
1.1.4 Dynamiques	7
1.1.5 Contraintes	9
1.2 Problèmes de contrôle optimal	11
1.2.1 Formalisation	11
1.2.2 Analyse	12
1.2.3 Problèmes simplifiés	13
1.3 État de l’art sur la modélisation	15
1.3.1 Architecture du véhicule	15
1.3.2 Modélisation	17
1.4 État de l’art sur les méthodes de résolution	17
1.4.1 Méthodes approchées	18
1.4.2 Méthodes basées sur la théorie du contrôle optimal	18
1.4.3 Méthodes à base d’intelligence artificielle	19
<b>Chapitre 2. Méthodes indirectes</b>	<b>21</b>
2.1 Problème de commande optimale et conditions nécessaires	22
2.1.1 Problème de commande optimale	22
2.1.2 Principe du maximum de Pontryagin	23
2.1.3 Hypothèse principale	26
2.2 Méthodes de résolution indirectes	30
2.2.1 Méthode de tir simple	30
2.2.2 Méthode de tir multiple	33
2.3 Parallélisation	33
2.3.1 Problème parallélisé	34
2.3.2 Lien avec le problème initial	35
2.3.3 Conditions nécessaires	36
2.3.4 Lien entre les conditions nécessaires d’optimalité	37
2.3.5 Schéma commutatif	40
2.3.6 Méthode numérique	40

<b>Chapitre 3. Méthode à deux niveaux</b>	<b>43</b>
3.1 Décomposition à deux niveaux	44
3.1.1 Formulation à deux niveaux	44
3.1.2 Lien avec le problème initial	47
3.1.3 Conditions nécessaires d'optimalité	51
3.1.4 Lien entre les conditions nécessaires d'optimalité	52
3.1.5 Graphe commutatif	56
3.2 Méthode numérique à deux niveaux	56
3.2.1 Idée principale	57
3.2.2 Les problèmes (Macro) et (Micro)	57
3.2.3 Structure du problème (Macro)	59
3.2.4 Analyse de l'erreur	60
3.2.5 Avantages et inconvénients	62
<b>Chapitre 4. Préconditionnement de la fonction de tir</b>	<b>65</b>
4.1 Formulation augmentée	66
4.1.1 Système augmenté	66
4.1.2 Conditions nécessaires d'optimalité	66
4.1.3 Ensemble accessible	67
4.2 Interprétation géométrique du co-état	69
4.2.1 Définitions et propriétés	69
4.2.2 Origine du co-état et cône de Boltyanski	72
4.2.3 Lien avec le cône tangent et le cône normal	73
4.2.4 Bilan	75
4.3 Préconditionnement géométrique	76
4.3.1 Principe	76
4.3.2 Cas 1 : l'ensemble accessible est difféomorphe à la boule	79
4.3.3 Cas 2 : extension au cas non difféomorphe	83
4.3.4 Cas général : utilisation de l'application exponentielle	89
<b>Chapitre 5. Expérimentations numériques</b>	<b>93</b>
5.1 Méthode de tir simple	94
5.1.1 Flot hamiltonien	94
5.1.2 Fonctions de tir	97
5.1.3 Méthode numérique	99
5.2 Méthode à deux niveaux	100
5.2.1 Construction d'approximations des fonctions valeur	101
5.2.2 Résolution	106
5.2.3 Résultats	108
5.2.4 Avantages	112
5.3 Préconditionnement géométrique	112
5.3.1 Preuve de concept	113
5.3.2 Généralisation	118

<b>Chapitre 6. Perspectives</b>	<b>123</b>
6.1 Multi-cycle . . . . .	124
6.1.1 Hypothèses . . . . .	124
6.1.2 Résolution . . . . .	124
6.2 Connaissance partielle . . . . .	125
6.2.1 Hypothèses . . . . .	126
6.2.2 Résolution . . . . .	127
6.3 Macro-Micro-ECMS . . . . .	127
6.3.1 Méthode ECMS . . . . .	128
6.3.2 Variante proposée . . . . .	128
<b>Chapitre 7. Conclusion</b>	<b>131</b>
<b>Chapitre A. Origine symplectique</b>	<b>135</b>
<b>Chapitre B. Preuves du chapitre 2</b>	<b>141</b>
<b>Chapitre C. Figures et tables</b>	<b>145</b>
<b>Bibliographie</b>	<b>153</b>





# Table des figures

---

1.1	Schéma véhicule hybride. . . . .	2
1.2	Trajectoire de vitesse du WLTC. . . . .	3
1.3	Architectures des HEV. . . . .	16
2.1	Fonctions de tir du problème (E). . . . .	32
2.2	Graphe commutatif des problèmes (OCP) et (POCP). . . . .	40
2.3	Graphe commutatif des problèmes (OCP) et (MPBVP)/(TPBVPP). . . . .	42
3.1	Graphe commutatif des problèmes (OCP) et (BOCP). . . . .	56
3.2	Graphe commutatif des problèmes (OCP), (POCP) et (BOCP). . . . .	57
3.3	Schéma de la méthode Macro-Micro. . . . .	58
4.1	Illustration de l'ensemble accessible. . . . .	68
4.2	Illustrations de la main de V. G. Boltyanski extraites de [7]. . . . .	73
4.3	Illustration d'ensembles accessibles convexe et non convexe. . . . .	74
4.4	Paramétrisation de l'ellipse. . . . .	81
4.5	Transformation de l'ellipse par $\phi$ . . . . .	83
4.6	Représentation schématique du préconditionnement, Cas 2. . . . .	86
4.7	Transformation de l'ensemble accessible par $\phi$ , Problème (E). . . . .	88
4.8	Fonctions de tir préconditionnées, Problème (E). . . . .	88
4.9	Représentation schématique du préconditionnement. . . . .	90
5.1	« Combat » des hamiltoniens. . . . .	95
5.2	Fonctions de tir. . . . .	97
5.3	Sortie du solveur. . . . .	99
5.4	Évolution du temps de calcul. . . . .	100
5.5	Schéma de la méthode Macro-Micro. . . . .	101
5.6	Points de la fonction valeur obtenus avec les deux méthodes. . . . .	104
5.7	Transformation par le flot hamiltonien. . . . .	105
5.8	Fonction valeur et son approximation. . . . .	106
5.9	Trajectoires d'état des méthodes de tir simple, Empirique et Macro-Micro. . . . .	111
5.10	Convergence du solveur. . . . .	111
5.11	Transformation de l'ensemble accessible par $\phi$ . . . . .	114
5.12	Représentation polaire, coordonnées initiales. . . . .	115
5.13	Représentation polaire, nouvelles coordonnées. . . . .	115
5.14	Évolution du co-état final en fonction du co-état initial. . . . .	116
5.15	Fonctions de tir préconditionnées. . . . .	117
5.16	Convergence du solveur avec préconditionnement. . . . .	118
5.17	Transformation de la base de données par $\phi_{\Delta}$ . . . . .	119
5.18	Transformation de l'ensemble accessible par $\phi_{\hat{x}_0}$ . . . . .	120
5.19	Fonctions de tir préconditionnées. . . . .	120

5.20	Convergence du solveur avec préconditionnement. ....	121
6.1	Schéma des méthodes proposées, multi-cycle. ....	125
6.2	Schéma des méthodes proposées, connaissance partielle. ....	127
6.3	Schéma de la méthode Macro-Micro-ECMS. ....	129
C.1	BC-extrémale. ....	147
C.2	Approximation l'ensemble des états admissibles. ....	148

# Liste des tableaux

---

1.1	Liste des entrées/sorties du modèle statique. ....	5
5.1	Description des réseaux de neurones. ....	105
5.2	Comparaison des solveurs pour le problème (Macro). ....	107
5.3	Coût du problème (Macro). ....	108
5.4	Comparaison des méthodes Macro-Micro et Empirique. ....	110
C.1	Liste des variables du modèle statique. ....	145
C.2	Liste des constantes du modèle statique. ....	146
C.3	Comparaison des solveurs du problème (Macro). ....	151



# Remerciements

---

Premièrement, je souhaite remercier les personnes qui ont rendu cette thèse possible, à savoir mon directeur de thèse Olivier Cots et mon directeur de thèse du monde socio-économique Mariano Sans. Vous m'avez permis d'une part de découvrir cette discipline si intéressante qui est la commande optimale, et d'autre part ce qu'est la recherche universitaire et industrielle. Je suis très reconnaissant pour la confiance que vous m'avez accordée, pour les connaissances que vous m'avez transmises et pour la grande pédagogie dont vous avez fait preuve.

Je remercie mes encadrants de thèse, à savoir Sophie Jan, Serge Laporte et Olivier Flebus. J'ai eu énormément de chance de vous avoir pendant ces trois années, pour vos connaissances, votre bienveillance et vos conseils. Vous avez été les référents que tout doctorant rêve d'avoir.

Je tiens à remercier Mme Hasnaa Zidani et M. Alain Rapaport pour avoir accepté d'être rapporteur de ce travail en tant qu'expert du contrôle optimal. Je remercie également M. Michel Povlovitsch Seixas et M. Jean-Baptiste Caillau pour avoir accepté d'être membre du jury de ma thèse. J'ai une pensée encore plus particulière pour vous, M. Caillau, qui m'avez transmis votre passion pour cette discipline dès vos cours en école d'ingénieur.

Merci à l'ensemble des membres de l'équipe APO, qui m'ont permis de passer ces trois années dans un cadre de recherche très agréable. Merci pour tous ces moments passés aux docteurs déjà partis : Boris, Bastien et Antoine ; à ceux restés : Théo et Sadok ; et à ceux qui le deviendront d'ici peu de temps : Sophie et Valentin.

Je remercie également tous les membres de T&I de Vitesco Technologies, qui m'ont si bien accueilli et intégré dans leur équipe. J'ai une pensée spéciale pour les membres fondateurs du club d'œnologie, Laurent et Maxime, aussi bien pour leur professionnalisme au travail que pour les très bons moments passés en dehors.

Je remercie mes amis de toujours : Nico, Mehdi, Raff et John. Malgré votre niveau très discuté sur la faille, vous restez des personnes importantes pour moi. Je remercie mes amis de prépa et d'école d'ingé : Gab, Momo, Alex et Mél. Merci pour votre soutien et pour tous ces moments inoubliables passés ensemble.

Enfin, je tiens à remercier ma famille. Vous m'avez toujours soutenu et encouragé à continuer, et si je suis là à écrire ces remerciements, c'est grâce à vous. Un immense merci à mes parents, qui ont fait de moi qui je suis.

J'ai une dernière pensée pour celle qui m'a supporté et soutenu durant ces deux dernières années. Ludivine, je t'embrasse fort.



# Introduction

---

**Application.** Les véhicules hybrides électriques utilisent deux sources d'énergie, respectivement le carburant et l'électricité stockée dans la batterie, et au moins deux moteurs, respectivement un moteur thermique et au moins un moteur électrique. Sur un parcours donné, en ayant un état de charge de la batterie initial et final fixés, le problème est de déterminer la répartition de couple entre ces deux moteurs ainsi que le choix du rapport de transmission de la boîte à vitesse qui minimise la consommation en carburant du véhicule. Ce problème industriel de commande optimale est bien connu dans la littérature, et de nombreuses méthodes, détaillées dans la section 1.4, ont été proposées afin d'obtenir des lois de commande efficaces. Nous nous intéressons dans ce document aux méthodes de contrôle optimal dites *indirectes*, c'est-à-dire basées sur le principe du maximum proposé par L. S. Pontryagin dans les années 1950 [43], qui doit sa formulation actuelle à R. V. Gamkrelidze et sa preuve à V. G. Boltyanski [7, 30]. Sous certaines hypothèses, ce principe mène à la résolution d'un problème au deux bouts, qui peut être résolu numériquement en cherchant un zéro de la fonction de tir simple à l'aide d'un solveur de type Newton. Comparées aux autres méthodes de commande optimale classiques (directes, basées sur les équations de Hamilton-Jacobi-Bellman, ...), les méthodes indirectes sont précises, rapides et utilisent peu de place mémoire, ce qui justifie leur choix pour une application qui a pour objectif d'être embarquée dans un véhicule. Elles sont cependant connues pour être sensibles à l'initialisation du solveur et demandent une connaissance a priori de la structure de la solution.

**Objectifs.** L'objectif principal de cette thèse est de proposer de nouvelles méthodes de commande optimale indirecte permettant de satisfaire des critères d'embarquabilité, c'est-à-dire qui soit

- **rapide.** La méthode proposée doit pouvoir fournir le contrôle en temps réel. Pour cela, on cherche à avoir la commande à appliquer au temps initial (ou actuel) le plus rapidement possible.
- **robuste.** Même si des garde-fous sont souvent utilisés dans ce type d'application, la méthode développée devra être assez robuste afin d'éviter au maximum d'y avoir recours. Nous considérons qu'une méthode indirecte est robuste si la fonction de tir associée est peu sensible et si une bonne initialisation peut être fournie au solveur de type Newton, rendant plus fiable la recherche de zéro.
- **légère.** La méthode proposée doit être implémentée dans un *Master Controller*, un module du véhicule produit par l'entreprise Vitesco Technologies. Elle ne doit donc pas être trop gourmande en place mémoire afin de ne pas rendre nécessaire un surdimensionnement de ce module, ce qui le rendrait trop coûteux.
- **optimale.** Nous voulons bien sûr proposer une solution optimale.



Il est difficile de satisfaire simultanément toutes ces exigences, aussi, nous avons choisi de proposer une méthode rapide, robuste et légère mais seulement approchée. Nous verrons toutefois qu'il est possible assez simplement d'en maîtriser le niveau d'approximation.

De plus, un autre objectif est de proposer des méthodes combinant des méthodes de contrôle optimal et des méthodes d'intelligence artificielles (IA). Plus précisément, on se place dans le cadre d'étude où l'IA aide à la résolution des méthodes de commande optimale, et non l'inverse.

**Démarche.** La méthode que nous avons développée a été élaborée dans le cadre industriel de l'entreprise Vitesco Technologies, où nous avons accès à un modèle industriel complexe qui représente finement les interactions entre les composants du véhicule hybride considéré. Développé en `Matlab Simulink`, il fait appel à des fonctions tabulées dont les données proviennent d'expérimentations réelles sur banc moteur. De plus, le parcours du véhicule correspond à un des cycles de roulage réel les plus utilisés dans l'industrie, à savoir le *Worldwide harmonized Light vehicles Test Cycle*, qui fait partie depuis 2019 de la procédure d'essai mondiale harmonisée pour les véhicules légers. De telles approches sont assez classiques pour ce type de problème [41, 44, 72]. Ce modèle ne fournit cependant pas d'expression explicite des dynamiques de notre problème, ce qui rend l'interprétation des solutions et la vérification des hypothèses considérées assez complexes.

Pour cette raison, une approche naturelle aurait été de développer un modèle plus simple où ces fonctions tabulées auraient été remplacées par des fonctions explicites. Nous aurions pu ensuite étudier ce problème simplifié dans le but d'obtenir une loi de commande optimale approchée. Enfin, la dernière étape aurait consisté à tester cette loi de commande sur le modèle initial afin d'en déterminer l'efficacité.

Ce n'est pourtant pas la démarche que nous avons choisie. En effet, nous avons préféré conserver le modèle initial sous forme de boîte noire afin que les méthodes développées dans ce document puissent être réutilisées par d'autres personnes du monde industriel et pour potentiellement d'autres problèmes de commande optimale semblables. Pour cette même raison, nous avons choisi d'utiliser les mêmes méthodes numériques que celles utilisées actuellement par l'entreprise Vitesco Technologies. De plus, les méthodes développées dans ce document sont présentées sous forme assez générale, où l'état et la commande sont de dimension arbitraire et où les contraintes initiales et finales d'état sont assez générales.

**Organisation du document.** Ce manuscrit est divisé en six chapitres relativement indépendants.

Le premier présente l'application considérée, c'est-à-dire le problème de répartition de couple dans un véhicule hybride électrique. Ce problème est écrit comme un problème de commande optimale sous la forme de Lagrange. La fin de ce chapitre est consacrée à un état de l'art des méthodes proposées dans la littérature pour sa résolution.

Le deuxième chapitre est décomposé en deux parties. La première introduit le cadre utilisé dans ce document et présente les méthodes indirectes de tir, à savoir le tir simple et multiple. La seconde partie donne un nouveau point de vue sur le tir multiple en l'interprétant comme l'application du tir simple à un problème parallélisé équivalent.

Au chapitre trois est proposée une nouvelle méthode de contrôle optimal : la méthode Macro-Micro. La première partie de ce chapitre est consacrée à l’origine de cette méthode, qui est basée sur une décomposition à deux niveaux du problème de commande optimale d’origine, et fait intervenir des fonctions valeur de Bellman à des temps fixés. La seconde partie de ce chapitre décrit la méthode Macro-Micro, où l’idée principale est d’approcher ces fonctions valeur, connues pour être difficiles à obtenir, par des réseaux de neurones.

Le quatrième chapitre propose une méthode de préconditionnement géométrique de la fonction de tir simple. La première partie de ce chapitre introduit deux notions essentielles pour la compréhension de ce préconditionnement, à savoir l’interprétation géométrique du co-état et la transformée de Mathieu. La seconde partie offre une construction progressive de ce préconditionneur, en commençant par un cas trivial pour aller jusqu’au cas général.

Les méthodes et les résultats numériques associés à la méthode de tir simple, à la méthode Macro-Micro et à la méthode de préconditionnement géométrique sur le problème de répartition de couple sont présentés dans le cinquième chapitre.

Finalement, le dernier chapitre propose des adaptations des méthodes proposées à des hypothèses moins fortes sur connaissance a priori du trajet, permettant de rendre ces méthodes compatibles aux exigences relatives à une solution embarquée.

**Collaborations.** L’entreprise Vitesco Technologies est à l’origine du financement de cette étude, effectuée entre l’équipe Technologie et Innovation (T&I) de la branche française de Vitesco Technologies, l’équipe Algorithmes Parallèles et Optimisation (APO) de l’Institut de Recherche en Informatique de Toulouse (IRIT) et de l’équipe Statistique et Optimisation (SO) de l’Institut de Mathématiques de Toulouse (IMT). De plus, cette thèse fait partie de la chaire *Artificial and Natural movement* de l’institut interdisciplinaire d’intelligence artificielle de Toulouse (ANITI pour *Artificial and Natural Intelligence Toulouse Institute*).

**Contributions.** Durant cette thèse, j’ai contribué au dépôt d’un brevet ainsi qu’à la rédaction d’un article et de deux proceedings. J’ai également présenté mes travaux dans trois conférences internationales. Ici sont données les références aux trois publications

- Olivier Cots, Rémy Dutto, Sophie Jan, Serge Laporte. *A bilevel optimal control method and application to the hybrid electric vehicle*, submitted, 2023.
- Olivier Cots, Rémy Dutto, Sophie Jan, Serge Laporte. *Generation of value function data for bilevel optimal control and application to hybrid electric vehicle*, in Thematic Einstein Semester conference on Mathematical Optimization for Machine Learning, 2023.
- Olivier Cots, Rémy Dutto, Sophie Jan, Serge Laporte. *Geometric preconditioner for indirect shooting and application to hybrid vehicle*, in 4th IFAC Conference of Modelling, Identification and Control of Nonlinear Systems, 2024.



# Application

---

<b>1.1</b>	<b>Description du problème</b>	<b>2</b>
1.1.1	Cycle	2
1.1.2	Relation entrée-sortie	5
1.1.3	Variables importantes	6
1.1.4	Dynamiques	7
1.1.5	Contraintes	9
<b>1.2</b>	<b>Problèmes de contrôle optimal</b>	<b>11</b>
1.2.1	Formalisation	11
1.2.2	Analyse	12
1.2.3	Problèmes simplifiés	13
<b>1.3</b>	<b>État de l'art sur la modélisation</b>	<b>15</b>
1.3.1	Architecture du véhicule	15
1.3.2	Modélisation	17
<b>1.4</b>	<b>État de l'art sur les méthodes de résolution</b>	<b>17</b>
1.4.1	Méthodes approchées	18
1.4.2	Méthodes basées sur la théorie du contrôle optimal	18
1.4.3	Méthodes à base d'intelligence artificielle	19

Dans ce chapitre est introduit le problème de répartition de couple et du rapport de transmission de la boîte à vitesses dans les véhicules hybrides électriques. Ce problème, assez étudié dans la littérature, a pour objectif de minimiser la consommation en carburant, tout en contrôlant l'état de charge de la batterie. Un autre état, correspondant à la température du liquide de refroidissement, est considéré dans la description du modèle. Nous allons cependant voir que sous des hypothèses assez simples, il est possible de se ramener à un problème de commande optimale à un seul état, étant l'état de charge de la batterie. Ce problème est considéré sur un des cycles les plus utilisés dans l'industrie, faisant partie de la procédure d'essai mondiale harmonisée pour les véhicules légers. L'objectif de ce chapitre est de formuler ce problème de commande optimale classique sous forme de Lagrange, et d'étudier les méthodes utilisées pour sa résolution.

## 1.1 Description du problème

On considère un véhicule hybride (HEV pour *Hybrid Electric Vehicle*), équipé d'un moteur électrique asynchrone de 400V (EM pour *Electric Motor*), d'un moteur thermique de 1,5L (ICE pour *Internal Combustion Engine*) et d'une batterie de 1.5 kWh. Le véhicule a une architecture « parallèle P3 », ce qui veut dire que l'arbre de transmission du moteur électrique est relié à l'arbre de transmission du moteur thermique après la boîte à vitesses thermique, comme montré dans la figure 1.1.

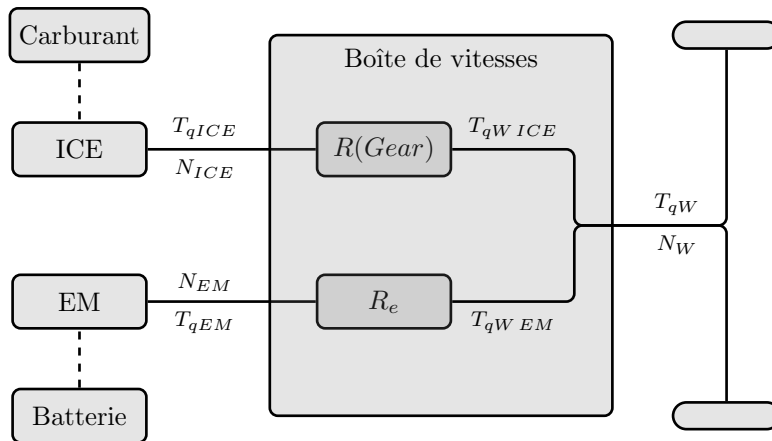


FIGURE 1.1 – Schéma du véhicule hybride considéré. Les lignes pleines correspondent aux connexions mécaniques, et les lignes discontinues aux connexions énergétiques (carburant pour l'ICE et électrique pour l'EM).

Le problème étudié dans cette thèse consiste à trouver la répartition de couple entre les moteurs électrique et thermique, ainsi que le choix du rapport de la boîte à vitesses, qui permettent de fournir le couple imposé aux roues et de minimiser la consommation de carburant, tout en ayant un état de charge de batterie initial et final fixé. L'objectif de cette section est de formuler ce problème comme un problème de commande optimale non autonome sous forme de Lagrange.

Pour cela, on commence par introduire la notion de cycle dans la section 1.1.1. La relation entrée-sortie du modèle statique est décrite dans la section 1.1.2, les variables intermédiaires importantes dans la section 1.1.3 ainsi que la dynamique d'état et de coût dans la section 1.1.4. Finalement, les contraintes du problème sont introduites dans la section 1.1.5.

### 1.1.1 Cycle

Un *cycle* correspond à une évolution de vitesse imposée à un véhicule sur une route dont la pente est connue. On peut donc résumer l'information d'un cycle à deux fonctions : la vitesse du véhicule  $v: [t_0, t_f] \rightarrow \mathbb{R}$  et la pente de la route  $p: [t_0, t_f] \rightarrow \mathbb{R}$ . On considère que ces fonctions sont suffisamment lisses. Un cycle est dit admissible pour un véhicule si celui-ci est capable d'avoir la vitesse  $v(t)$  sur une route de pente  $p(t)$ , pour tout temps  $t \in [t_0, t_f]$ .

**Remarque.** On considère que la vitesse de la voiture  $v(t)$  est positive, pour tout temps  $t \in [t_0, t_f]$ . On ne considère donc pas le scénario de marche arrière.

Parmi les cycles admissibles pour notre véhicule, nous avons sélectionné le WLTC (pour *Worldwide harmonized Light vehicles Test Cycle*). Ce cycle fait partie de la procédure d'essai mondiale harmonisée pour les véhicules légers, qui permet de comparer la consommation en carburant et l'émission de gaz de différents véhicules. Ce cycle se veut représentatif des vitesse et accélération des véhicules, ainsi que des manières de conduire des conducteurs. Le WLTC est composé d'une trajectoire de vitesse de 1800 secondes montrée dans la figure 1.2 et d'une pente nulle pour tous les temps.

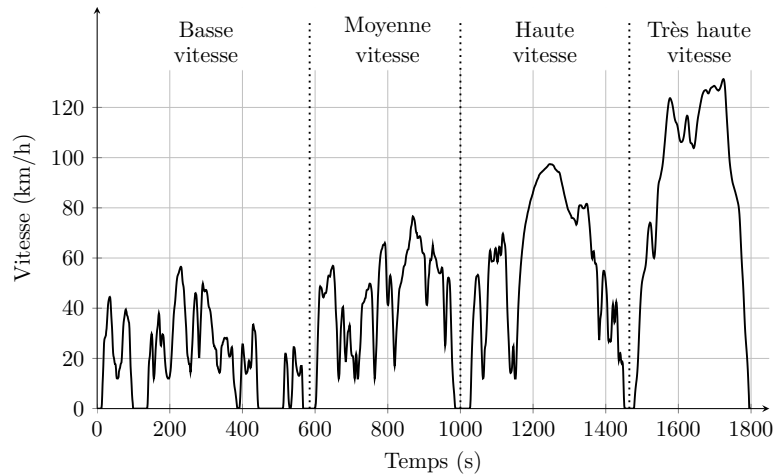


FIGURE 1.2 – Trajectoire de vitesse du WLTC.

Le cycle fixe le temps initial et final. Dans notre cas, nous avons donc  $t_0 = 0$  et  $t_f = 1800$ . De plus, il est possible d'obtenir la vitesse de rotation des roues  $N_W(t)$  et le couple demandé aux roues  $T_{qW}(t)$  pour tout temps  $t \in [t_0, t_f]$ . Ces deux séries temporelles sont des entrées du modèle de notre véhicule.

**Vitesse de rotation des roues.** La vitesse de rotation des roues  $N_W(t)$  se calcule à partir de la vitesse du véhicule  $v(t)$  de la manière suivante

$$N_W(t) = \frac{60}{2\pi r} v(t)$$

où  $r$  correspond au rayon des roues de la voiture.

**Couple demandé aux roues.** Pour calculer le couple demandé aux roues, on calcule d'abord l'accélération  $a(t)$  de la voiture en dérivant la vitesse

$$a(t) = \dot{v}(t).$$

On utilise ensuite la deuxième loi de Newton

$$a(t) = \frac{F_T(t)}{m}$$

où  $m$  est la masse de la voiture, considérée constante, et  $F_T(t)$  correspond à la force totale appliquée à la voiture.

**Remarque.** Dans ce chapitre, nous appelons « force » le module de la projection, sur l'axe longitudinal de la voiture, du vecteur force correspondant.

Cette force totale est calculée par

$$F_T(t) = F_r(t) + F_p(t) + F_d(t)$$

où

- $F_r(t)$  correspond aux forces de résistance aérodynamique, modélisées par un polynôme de degré 2 en la vitesse, si celle-ci est positive :

$$F_r(t) = \begin{cases} -(F_0 + F_1 \cdot v(t) + F_2 \cdot v^2(t)) & \text{si } v(t) > 0, \\ 0 & \text{sinon ;} \end{cases}$$

- $F_p(t)$  correspond à la force de pesanteur, qui dépend de la pente de la route. Cette force est calculée par

$$F_p(t) = -m \cdot g \cdot \sin(\theta(t)),$$

où  $g = 9.80665 \text{ m.s}^{-2}$  correspond à l'accélération de la pesanteur terrestre, et  $\theta(t)$  à l'angle entre la route de pente  $p(t)$  et l'horizontale

$$\theta(t) = \arctan(p(t)) ;$$

- $F_d(t)$  correspond à la force provenant de la propulsion du véhicule, calculée par

$$F_d(t) = \frac{1}{r} (T_{qW}(t) - T_{qF}(t))$$

avec  $T_{qF}(t)$  le couple utilisé par les freins mécaniques. Ce couple positif correspond à la différence entre le couple de freinage demandé aux roues et le couple maximum que peut absorber le moteur électrique, et est obtenu par une simulation préliminaire sur la totalité du cycle.

En utilisant les équations précédentes, il est possible de calculer  $T_{qW}(t)$  à partir des informations du cycle au temps  $t$ , c'est-à-dire la vitesse du véhicule  $v(t)$  et la pente de la route  $p(t)$ . Finalement, les deux informations issues du cycle utiles pour notre problème de répartition de couple sont résumées par le couple aux roues  $T_{qW}(\cdot)$  et la vitesse de rotation des roues  $N_W(\cdot)$ , considérés lisses.

On suppose dans la suite que l'on a une connaissance totale et parfaite des fonctions  $T_{qW}$  et  $N_W$ . Cette hypothèse, bien que très forte, est déjà utilisée dans la plupart des références de la littérature. Elle sera remise en question dans le chapitre 6. Pour la description du modèle statique, c'est-à-dire lorsque le temps  $t \in [t_0, t_f]$  est fixé, on s'autorise à ne plus écrire la dépendance en temps des variables, afin de simplifier les notations.

### 1.1.2 Relation entrée-sortie

L'objectif de cette partie est de décrire la relation entrée-sortie du modèle statique. Les entrées de ce modèle sont

- les deux variables introduites précédemment ( $N_W$  et  $T_{qW}$ ) qui sont ici vues comme des perturbations,
- les deux états : l'état de charge de la batterie  $SOC$  et la température du liquide de refroidissement  $T$ ,
- les deux commandes : le rapport de la boîte à vitesses  $Gear$  et le couple du moteur thermique  $T_{qICE}$ .

La consommation instantanée en carburant  $\dot{m}_F$ , la variation instantanée en état de charge de batterie  $\dot{SOC}$  ainsi que la variation instantanée de la température du liquide de refroidissement  $\dot{T}$  forment les sorties du modèle.

La liste des entrées, sorties et variables intermédiaires importantes du modèle statique est donnée dans la table 1.1. Les autres variables intermédiaires et constantes utilisées pour la description du modèle sont listées dans les tables C.1 et C.2 en annexe.

Variable	Description	Unité
États		
$SOC$	État de charge de la batterie	sans dimension
$T$	Température de l'ICE	°C
Commandes		
$T_{qICE}$	Couple de l'ICE	N.m
$Gear$	Rapport de la boîte à vitesses	sans dimension
Perturbations		
$N_W$	Vitesse de rotation des roues	RPM <sup>1</sup>
$T_{qW}$	Couple demandé aux roues	N.m
Sorties		
$\dot{m}_F$	Débit massique de carburant	g.s <sup>-1</sup>
$\dot{SOC}$	Variation instantanée de la $SOC$	s <sup>-1</sup>
$\dot{T}$	Variation instantanée de $T$	°C.s <sup>-1</sup>
Variables intermédiaires importantes		
$N_{EM}$	Vitesse de rotation de l'EM	RPM
$N_{ICE}$	Vitesse de rotation de l'ICE	RPM
$T_{qEM}$	Couple de l'EM	N.m
$PME$	Pression moyenne effective de l'ICE	Pa

TABLE 1.1 – Liste des entrées, sorties et variables intermédiaires importantes du modèle statique.

---

1. Rotations Par Minute



### 1.1.3 Variables importantes

Les variables intermédiaires importantes sont celles qui apparaissent dans le calcul d'au moins deux sorties du modèle statique. On propose donc de les décrire maintenant.

**Vitesse de rotation du moteur électrique.** La vitesse de rotation du moteur électrique (et de son arbre)  $N_{EM}$  se calcule à l'aide de la vitesse de rotation des roues  $N_W$ . En effet, l'axe du moteur électrique est toujours connecté à l'axe principal (qui tourne à la vitesse des roues) à travers le système de transmission. On a donc

$$N_{EM} = R_e N_W$$

où  $R_e$  correspond au rapport de transmission entre l'arbre principal et l'arbre du moteur électrique. On a donc

$$N_{EM} = f_{N_{EM}}(N_W).$$

**Vitesse de rotation du moteur Thermique.** La vitesse de rotation du moteur thermique (et de son arbre)  $N_{ICE}$  est calculée à partir de la vitesse de rotation de l'arbre principal  $N_W$  ainsi que de la valeur du  $Gear$ . En effet, le moteur thermique est connecté à l'arbre principal en passant par la boîte à vitesses mécanique. On a donc

$$N_{ICE} = R(Gear)N_W$$

où  $R: \mathbb{R} \rightarrow \mathbb{R}$  est considéré comme une fonction de classe  $\mathcal{C}^\infty$  et correspond à la fonction qui fournit le ratio de transmission entre l'arbre principal et l'arbre du moteur thermique.

**Remarque.** La fonction  $R$  est en réalité une fonction discrète, définie sur  $\{0, 1, 2, 3, 4\}$ . Il est cependant possible de la construire comme une fonction de classe  $\mathcal{C}^\infty$  définie sur  $\mathbb{R}$  (un polynôme de degré 4 par exemple), et de faire porter la contrainte sur la commande  $Gear \in \{0, 1, 2, 3, 4\}$ . De plus, si  $Gear = 0$ , alors l'arbre du moteur thermique n'est plus connecté à l'arbre principal. Cette propriété implique que  $R(0) = 0$ .

On a donc

$$N_{ICE} = f_{N_{ICE}}(N_W, Gear).$$

**Couple du moteur électrique.** Le couple du moteur électrique  $T_{qEM}$  peut être calculé à partir du bilan des couples au niveau de la transmission

$$T_{qW} = T_{qWICE} + T_{qWEM},$$

où  $T_{qWICE}$  (resp.  $T_{qWEM}$ ) correspond à la part de  $T_{qW}$  générée aux roues par l'ICE (resp. EM). De leur côté,  $T_{qWICE}$  et  $T_{qWEM}$  s'écrivent sous la forme

$$T_{qWICE} = T_{qWICE}^* - T_{qLICE},$$

$$T_{qWEM} = T_{qWEM}^* - T_{qLEM},$$

où  $T_{qWICE}^*$  et  $T_{qWEM}^*$  correspondent respectivement aux couples que fournirait le moteur thermique (respectivement électrique) si les pertes de la transmission  $T_{qLICE}$  et  $T_{qLEM}$  étaient négligées. Ces couples idéaux se calculent par

$$\begin{aligned} T_{qWICE}^* &= R(Gear)T_{qICE}, \\ T_{qWEM}^* &= R_e T_{qEM}. \end{aligned}$$

De leur côté, les pertes de la boîte à vitesses peuvent être calculées de deux façons chacune, à l'aide des fonctions tabulées suivantes

$$\begin{aligned} T_{qLICE} &= M_{ICE1}(N_{ICE}, T_{qICE}) \quad \text{ou} \quad T_{qLICE} = M_{ICE2}(N_W, T_{qWICE}), \\ T_{qLEM} &= M_{EM1}(N_{EM}, T_{qEM}) \quad \text{ou} \quad T_{qLEM} = M_{EM2}(N_W, T_{qWEM}). \end{aligned}$$

Finalement, on a

$$\begin{aligned} T_{qEM} &= \frac{1}{R_e} (T_{qWEM} + T_{qLEM}) \\ &= \frac{1}{R_e} \left( T_{qW} - T_{qWICE} + M_{EM2}(N_W, T_{qW} - T_{qWICE}) \right). \end{aligned}$$

Comme

$$T_{qWICE} = R(Gear)T_{qICE} - M_{ICE1}(N_{ICE}, T_{qICE}),$$

étant donné les relations déjà établies pour  $N_{EM}$  et  $N_{ICE}$ , on aboutit à

$$T_{qEM} = f_{T_{qEM}}(T_{qW}, N_W, T_{qICE}, Gear).$$

**Pression effective moyenne du moteur thermique.** La pression moyenne effective au sein du moteur thermique  $PME$  est calculée à partir du couple idéal du moteur thermique  $T_{qiICE}$  par

$$PME = P.T_{qiICE}$$

où  $P$  est un coefficient qui permet de convertir le couple en pression. Le couple indiqué du moteur thermique  $T_{qiICE}$  correspond au couple fourni par le moteur thermique quand les pertes sont négligées, c'est-à-dire

$$T_{qiICE} = T_{qICE} + T_{qfICE}.$$

Les pertes de couple du moteur thermique  $T_{qfICE}$  sont principalement des pertes liées aux frottements et sont calculées à partir de la fonction tabulée suivante

$$T_{qfICE} = M_{T_{qfICE}}(N_{ICE}, T).$$

On a donc

$$PME = f_{PME}(N_W, T, T_{qICE}, Gear).$$

#### 1.1.4 Dynamiques

Nous venons de décrire comment les variables intermédiaires importantes sont construites, et nous allons maintenant décrire les relations entrées-sorties du modèle statique, c'est-à-dire décrire comment sont calculées  $\dot{m}_F$ ,  $S\dot{O}C$  et  $\dot{T}$  en fonction des entrées du modèle statique.

**Consommation en carburant.** La consommation instantanée en masse de carburant  $\dot{m}_F$  est calculée directement à l'aide de la fonction tabulée suivante

$$\dot{m}_F = M_{\dot{m}_F}(N_{ICE}, PME).$$

On a donc

$$\dot{m}_F = f_{\dot{m}_F}(N_W, T, T_{qICE}, Gear).$$

**État de charge de la batterie.** La variation instantanée de l'état de charge de batterie  $S\dot{O}C$  se calcule à partir de l'intensité  $I$  du courant dans la batterie de la manière suivante

$$S\dot{O}C = \frac{I}{Q}$$

avec  $Q$  la capacité équivalente de la batterie.

**Remarque.** Nous avons pris la convention de considérer que si  $I < 0$  alors le moteur électrique fournit du couple à l'arbre du moteur électrique, et donc aux roues. A contrario, si  $I > 0$ , alors c'est l'arbre du moteur électrique qui fournit du couple au moteur électrique. Dans ce cas, ce couple provenant des roues et/ou du moteur thermique permet de recharger la batterie.

L'intensité  $I$  du courant est calculée à partir de la fonction tabulée suivante

$$I = M_I(SOC, P_{eT})$$

où  $P_{eT}$  correspond à la puissance électrique totale utilisée par le véhicule. Cette puissance correspond à la somme de la puissance utilisée par le moteur électrique  $P_{EM}$  et de la puissance additionnelle moyenne des autres équipements de la voiture  $P_{add}$  considérée constante

$$P_{eT} = P_{EM} + P_{add}.$$

La puissance utilisée par le moteur électrique  $P_{EM}$  est calculée à partir de la fonction tabulée suivante

$$P_{EM} = M_{P_{EM}}(N_{EM}, T_{qEM}).$$

On a donc

$$S\dot{O}C = f_{S\dot{O}C}(T_{qW}, N_W, SOC, T_{qICE}, Gear).$$

**Température du liquide de refroidissement.** La variation instantanée de température du liquide de refroidissement est calculée à partir de la puissance thermique totale  $P_T$  :

$$\dot{T} = \frac{P_T}{C}$$

où  $C$  correspond à la capacité thermique du liquide de refroidissement. La puissance thermique totale est calculée à partir du bilan des puissances thermiques, c'est-à-dire

$$P_T = P_f + P_c - P_a - P_t - P_h$$

avec :

- $P_f$  la puissance calorifique due à la friction du moteur thermique, calculée par

$$P_f = AT_{qICE} + M_{P_f}(N_{ICE}, T)$$

où  $A$  est un coefficient de modélisation et  $M_{P_f}$  une fonction tabulée ;

- $P_c$  la puissance calorifique dégagée par la combustion, calculée par la fonction tabulée suivante

$$P_c = M_{P_c}(N_{ICE}, PME);$$

- $P_a$  la puissance transférée à l'air ambiant extérieur, calculée par

$$P_a = D(T - T_a)$$

où  $D$  correspond au coefficient de transfert thermique entre l'air et le liquide de refroidissement de l'ICE, et  $T_a$  est la température de l'air extérieur, considérée constante ;

- $P_t$  la puissance calorifique dégagée par le thermostat, calculée par

$$P_t = F \text{ReLU}(T - 90) \quad (1.1)$$

où  $F$  correspond au coefficient d'efficacité du thermostat, et où la fonction unité linéaire rectifiée (*Rectified Linear Unit*)  $\text{ReLU}: \mathbb{R} \rightarrow \mathbb{R}$  est définie par

$$\text{ReLU}(x) = \max(0, x).$$

On remarque que le thermostat est actif uniquement lorsque la température de l'ICE dépasse les 90°C ;

- $P_h$  la puissance calorifique moyenne utilisée pour chauffer l'habitacle, considérée constante et connue. Cette simplification a été validée par l'un des constructeurs automobiles consultés par Vitesco Technologies dans le cadre d'un projet connexe à cette thèse.

On a donc

$$\dot{T} = f_{\dot{T}}(N_W, T, T_{qICE}, Gear).$$

Finalement, nous venons de décrire dans cette section la dynamique du coût ( $\dot{m}_F$ ) et la dynamique d'état ( $\dot{SOC}$  et  $\dot{T}$ ) à partir des entrées de notre modèle, à savoir les perturbations ( $N_W$  et  $T_{qW}$ ), l'état ( $SOC$  et  $T$ ) et la commande ( $T_{qICE}$  et  $Gear$ ).

### 1.1.5 Contraintes

Il existe des contraintes de commande et d'état dans le problème de répartition de couple, qui sont principalement dues à des limitations physiques (limites de vitesse de rotation des moteurs,  $Gear$  discret, limites d'état de charge de batterie) mais également au confort des passagers. Nous allons maintenant les définir. Pour cela, on reste dans le cadre de la description du modèle statique, c'est-à-dire qu'on s'autorise toujours à ne pas indiquer la dépendance en temps des variables.

**La boîte à vitesses.** La boîte à vitesses automatique considérée est une boîte à simple embrayage, à quatre rapports. On a donc

$$Gear \in \{0, 1, 2, 3, 4\} =: U_{Gear}.$$

**Le couple du moteur thermique.** Les contraintes sur le couple du moteur thermique, pour une valeur de  $Gear$  fixée, s'expriment sous la forme

$$T_{qICE} \in \{T_{qICE}^-\} \cup [0, T_{qICE}^+]$$

où  $T_{qICE}^-$  (respectivement  $T_{qICE}^+$ ) correspond à la borne minimale négative ou nulle (respectivement maximale positive ou nulle) admissible pour le couple du moteur thermique. Ces deux bornes sont calculées à l'aide des fonctions tabulées suivantes

$$T_{qICE}^- = M_{T_{qICE}^-}(N_{ICE}) \quad \text{et} \quad T_{qICE}^+ = M_{T_{qICE}^+}(N_{ICE}).$$

**Remarque.** À vitesse fixée, le couple du moteur thermique peut être négatif, mais ne peut prendre qu'une seule valeur lorsque le  $Gear$  est fixé. C'est ce que l'on appelle plus communément le frein moteur.

On a donc

$$T_{qICE} \in U_{T_{qICE}}(N_W, Gear).$$

**Le couple du moteur électrique.** Le moteur électrique ne fonctionne pas comme le moteur thermique. En effet, le couple du moteur électrique peut prendre l'ensemble des valeurs entre son minimum et son maximum. Nous avons donc

$$T_{qEM} \in [T_{qEM}^-, T_{qEM}^+],$$

dont les bornes sont calculées à l'aide des fonctions tabulées suivantes

$$T_{qEM}^- = M_{T_{qEM}^-}(N_{EM}) \quad \text{et} \quad T_{qEM}^+ = M_{T_{qEM}^+}(N_{EM}).$$

On a donc

$$T_{qEM} \in U_{T_{qEM}}(N_W).$$

**La vitesse de rotation du moteur thermique.** Du fait de contraintes mécaniques, la vitesse de rotation du moteur thermique est comprise entre une valeur minimale (appelée régime de ralenti)  $N_{ICE}^-$  et une valeur maximale  $N_{ICE}^+$ , c'est-à-dire

$$N_{ICE}^- \leq N_{ICE} \leq N_{ICE}^+.$$

De plus, pour le confort des passagers du véhicule, le moteur thermique n'est pas en fonctionnement si la vitesse du véhicule est trop faible, c'est-à-dire

$$N_{ICE} = \begin{cases} 0 & \text{si } N_W < N_W^-, \\ f_{N_{ICE}}(N_W, Gear) & \text{sinon,} \end{cases}$$

où  $N_W^-$  correspond à la vitesse minimale de rotation des roues qui autorise l'utilisation du moteur thermique. On a donc

$$N_{ICE} \in U_{N_{ICE}}(N_W).$$

**L'état de charge de la batterie.** L'état de charge de la batterie est par définition un scalaire compris entre zéro et un. Cependant, pour éviter une dégradation trop précoce de la batterie chimique, on impose

$$0.2 \leq SOC \leq 0.8. \quad (1.2)$$

On a donc

$$SOC \in [0.2, 0.8].$$

**Remarque.** Il est possible de considérer également des contraintes d'état sur la température du liquide de refroidissement. Nous allons cependant considérer que ces contraintes ne sont pas nécessaires.

## 1.2 Problèmes de contrôle optimal

À partir de cette section, on ne s'intéresse plus au modèle statique, mais au modèle dynamique. Les états et les commandes ne sont donc plus des scalaires, mais des fonctions de  $[t_0, t_f]$  à valeurs dans  $\mathbb{R}$ . Plus précisément, l'état  $x \in AC([t_0, t_f], \mathbb{R}^2)$  est une fonction absolument continue définie sur l'intervalle de temps  $[t_0, t_f]$  à valeurs dans  $\mathbb{R}^2$  et correspond aux trajectoires de l'état de charge de batterie et de la température du liquide de refroidissement du moteur thermique. La commande  $u \in L^\infty([t_0, t_f], \mathbb{R}^2)$  est une fonction essentiellement bornée définie sur l'intervalle  $[t_0, t_f]$  à valeurs dans  $\mathbb{R}^2$  et correspond aux trajectoires du couple du moteur thermique et du rapport de boîte *Gear*.

L'objectif est maintenant d'étudier le problème de répartition de couple sous la forme d'un problème de commande optimale classique, et dont la résolution peut être effectuée classiquement par des méthodes indirectes. Pour cela, nous allons dans la section 1.2.1 écrire le problème de commande optimale sous une forme plus compacte, le problème (P<sub>1</sub>), pour ensuite analyser les propriétés de ce problème dans la section 1.2.2. La section 1.2.3 propose enfin deux versions simplifiées (P<sub>2</sub>) et (P<sub>3</sub>) de (P<sub>1</sub>). Le problème (P<sub>3</sub>) sera utilisé pour les expérimentations numériques dans le chapitre 5.

### 1.2.1 Formalisation

Le problème de répartition de couple peut être formulé de la façon suivante

$$(P_1) \quad \begin{cases} \min_{x,u} \int_{t_0}^{t_f} f^0(t, x(t), u(t)) dt \\ \text{s.c. } \dot{x}(t) = f(t, x(t), u(t)), & t \in [t_0, t_f] \text{ p.p.}, \\ u(t) \in U(t), & t \in [t_0, t_f], \\ x(t) \in \mathcal{X}, & t \in [t_0, t_f], \\ c(x(t_0), x(t_f)) = 0 \end{cases}$$

où, en notant  $x = (SOC, T) \in \mathbb{R}^2$  et  $u = (T_{qICE}, Gear) \in \mathbb{R}^2$  :

- la dynamique du coût  $f^0: \mathbb{R} \times \mathbb{R}^2 \times \mathbb{R}^2 \rightarrow \mathbb{R}$  est définie par

$$f^0(t, x, u) = f_{m_F}(N_W(t), T, T_{qICE}, Gear),$$

- la dynamique d'état  $f: \mathbb{R} \times \mathbb{R}^2 \times \mathbb{R}^2 \mapsto \mathbb{R}^2$  est définie par

$$f(t, x, u) = \begin{pmatrix} f_{SOC}(T_{qW}(t), N_W(t), SOC, T_{qICE}, Gear) \\ f_{\dot{T}}(N_W(t), T, T_{qICE}, Gear) \end{pmatrix},$$

- le domaine d'état  $\mathcal{X} \subset \mathbb{R}^2$  est défini par

$$\mathcal{X} = [0.2, 0.8] \times \mathbb{R},$$

- pour tout  $t \in [t_0, t_f]$ , le domaine de commande  $U(t) \subset \mathbb{R}^2$  est l'ensemble fermé non vide défini par

$$\left\{ (T_{qICE}, Gear) \left| \begin{array}{l} T_{qICE} \in U_{T_{qICE}}(N_W(t), Gear), \\ Gear \in U_{Gear}, \\ f_{T_{qEM}}(T_{qW}(t), N_W(t), T_{qICE}, Gear) \in U_{T_{qEM}}(N_W(t)), \\ f_{N_{ICE}}(N_W(t), Gear) \in U_{N_{ICE}}(N_W(t)). \end{array} \right. \right\},$$

- la fonction de contrainte sur les états initiaux et finaux  $c: \mathbb{R}^2 \times \mathbb{R}^2 \mapsto \mathbb{R}^3$  est définie par

$$c((a_1, a_2), (b_1, b_2)) = \begin{pmatrix} a_1 - SOC_0 \\ a_2 - T_0 \\ b_1 - SOC_T \end{pmatrix}$$

avec  $SOC_0, T_0, SOC_T \in \mathbb{R}$  fixés.

Il est possible de formuler l'ensemble  $U(t)$  de manière plus simple par

$$U(t) = \left\{ (T_{qICE}, Gear) \left| \begin{array}{l} Gear \in U_1(t), \\ T_{qICE}(t) \in U_2(t, Gear) \end{array} \right. \right\}. \quad (1.3)$$

Pour tout  $t \in [t_0, t_f]$ , l'ensemble  $U_1(t)$  correspond à un ensemble discret

$$U_1(t) \subset \{0, 1, 2, 3, 4\},$$

et pour tout  $Gear \in U_1(t)$ , l'ensemble  $U_2(t, Gear)$  peut s'écrire de la manière suivante

$$U_2(t, Gear) = \{T_-(t, Gear)\} \cup [T_+^-(t, Gear), T_+^+(t, Gear)],$$

où  $T_-(t, Gear)$  est une fonction à valeur négative, et correspond d'un point de vue physique à l'utilisation du frein moteur. Les fonctions  $T_+^-(t, Gear)$  et  $T_+^+(t, Gear)$  correspondent respectivement au couple minimum et maximum autorisé du moteur thermique, en prenant en compte les limitations de couple et de vitesse de rotation des moteurs électrique et thermique.

### 1.2.2 Analyse

Le problème ( $P_1$ ) est un problème de commande optimale non autonome sous forme de Lagrange avec des contraintes d'état et de commande. L'objectif de cette section est d'analyser les propriétés de ce problème.

Commençons par discuter de la régularité des fonctions de dynamique et de coût en considérant l'hypothèse suivante.

**Hypothèse H1** (Régularité des dynamiques).

Les fonctions tabulées utilisées pour modéliser les dynamiques d'états, la dynamique de coût, ainsi que les contraintes sont des fonctions lisses, c'est-à-dire au moins  $\mathcal{C}^1$ .

**Remarque.** Ces fonctions tabulées sont basées sur des données calculées sur banc moteur. Elles appartiennent à l'entreprise Vitesco Technologies et ne peuvent donc pas être partagés pour cause de confidentialité.

Sous cette hypothèse, on remarque que  $f^0$  est  $\mathcal{C}^1$  en état et commande, et la fonction de dynamique d'état  $f$  est  $\mathcal{C}^1$  en commande. Cependant, en raison de l'équation (1.1), la fonction  $f$  est seulement Lipschitz continue en état. En définissant deux strates d'état  $\Omega_1$  et  $\Omega_2$  de la manière suivante

$$\Omega_1 = \{x \in \mathcal{X} \mid x_2 \leq 90\} \quad \text{et} \quad \Omega_2 = \{x \in \mathcal{X} \mid x_2 > 90\},$$

on remarque alors que  $f$  peut se mettre sous la forme

$$f(t, x, u) = \begin{cases} f_t(t, x, u) & \text{si } x \in \Omega_1, \\ f_p(t, x, u) & \text{si } x \in \Omega_2, \end{cases}$$

où  $f_t$  et  $f_p$  sont deux fonctions  $\mathcal{C}^1$  en état et commande. La fonction  $f_t$  correspond à la dynamique d'état en régime transitoire, c'est-à-dire lorsque le moteur thermique est froid et monte en température. Pour sa part,  $f_p$  correspond à la dynamique en régime permanent, c'est-à-dire lorsque le moteur thermique est chaud et que la température du liquide de refroidissement est proche des 90 °C. Le problème (P<sub>1</sub>) est donc un problème de commande optimale non lisse [19], ce qui peut être également vu de manière plus générale comme un problème de commande optimale hybride [9, 27]. Ce problème est assez complexe, dans le sens où il est assez compliqué de connaître à l'avance le nombre de transitions entre les deux strates à la solution.

### 1.2.3 Problèmes simplifiés

On propose de construire un problème simplifié proche du problème initial en travaillant sur deux aspects. Premièrement, on propose d'ignorer la contrainte d'état donnée par  $\mathcal{X}$  et de vérifier a posteriori que la trajectoire optimale satisfait cette contrainte.

**Remarque.** Dans l'ensemble des expériences effectuées, nous n'avons pas eu de cas où la solution ne satisfait pas cette contrainte. Il est cependant possible de construire un problème, avec une condition initiale et finale en état de charge de batterie proche d'une de ces bornes, qui ne satisfait pas cette contrainte si celle-ci est ignorée. Nous ne nous occupons pas de ces cas dans ce document.

On propose également de simplifier la dynamique de température, dont la complexité provient de l'équation (1.1). Une méthode simple serait de régulariser la fonction ReLU qui intervient dans cette équation en l'approchant par une fonction lisse. Cependant, comme cette singularité traduit un phénomène physique d'activation et désactivation du thermostat, nous proposons plutôt de nous ramener à un problème de commande optimale où il est plus simple de connaître à l'avance le nombre et l'ordre des transitions entre les deux strates à la solution. Pour ce faire, on suppose que le thermostat de la



voiture est bien dimensionné (c'est-à-dire que  $F$  est assez grand) et que le moteur thermique est utilisé suffisamment souvent à la solution pour que la température du liquide de refroidissement ne descende pas trop. Sous les deux hypothèses précédentes, et en notant

$$t_1 = \min \left\{ \{t \mid T(t) = 90\} \cup \{t_f\} \right\},$$

il existe  $\varepsilon \geq 0$  assez petit tels que pour tout  $t \in ]t_1, t_f[$ ,

$$-\varepsilon \leq T(t) - 90 \leq \varepsilon.$$

En effet, si le thermostat est bien dimensionné, il dissipe bien la chaleur, ce qui empêche la température du liquide de refroidissement d'aller trop au-delà de 90 °C, et l'utilisation fréquente du moteur thermique à la solution empêche la température du liquide de refroidissement de descendre trop sous les 90 °C. De plus, on suppose que pour toute température  $T \in [90 - \varepsilon, 90 + \varepsilon]$ , pour toute commande  $u = (T_{qICE}, Gear) \in \mathbb{R}^2$  et pour tout temps  $t \in \mathbb{R}$ , on a

$$f_{m_F}(N_W, T, T_{qICE}, Gear) \approx f_{m_F}(N_W, 90, T_{qICE}, Gear).$$

On peut donc considérer que dès que la température du liquide de refroidissement a atteint les 90°C, celle-ci est ensuite constante et égale à 90°C. Le problème de commande optimale associé s'écrit donc

$$(P_2) \quad \begin{cases} \min_{x,u} \int_{t_0}^{t_f} f^0(t, x(t), u(t)) dt \\ \text{s.c. } \dot{x}(t) = f(t, x(t), u(t)), & t \in [t_0, t_f] \text{ p.p.}, \\ u(t) \in U(t), & t \in [t_0, t_f], \\ c(x(t_0), x(t_f)) = 0 \end{cases}$$

où la dynamique d'état  $f: \mathbb{R} \times \mathbb{R}^2 \times \mathbb{R}^2 \rightarrow \mathbb{R}^2$  est redéfinie par

$$f(t, x, u) = \begin{cases} f_t(t, x, u) & \text{si } x_2 < 90, \\ (f_{t,1}(t, x, u), 0) & \text{si } x_2 = 90. \end{cases}$$

On remarque qu'il est impossible de sortir de la strate  $\Omega_2$  sous les hypothèses faites ci-dessus. Il existe donc trois combinaisons possibles des strates :

$$\begin{cases} \Omega_1 & \text{si } T(t_0) < 90^\circ\text{C et } T(t_f) < 90^\circ\text{C}, \\ \Omega_1 \text{ puis } \Omega_2 & \text{si } T(t_0) < 90^\circ\text{C et } T(t_f) = 90^\circ\text{C}, \\ \Omega_2 & \text{si } T(t_0) = 90^\circ\text{C}. \end{cases}$$

Nous allons nous concentrer dans la suite sur la résolution du problème de répartition de couple et du choix du *Gear* en régime permanent, c'est-à-dire lorsque  $T_0 = T(t_0) = 90^\circ\text{C}$ . Dans ce cas, le problème de commande optimale  $(P_2)$  est un problème de commande optimale à un seul état (l'état de charge de la batterie). Ce problème peut être écrit de

la manière suivante

$$(P_3) \quad \begin{cases} \min_{x,u} \int_{t_0}^{t_f} f^0(t, u(t)) dt \\ \text{s.c. } \dot{x}(t) = f(t, x(t), u(t)), & t \in [t_0, t_f] \text{ p.p.}, \\ u(t) \in U(t), & t \in [t_0, t_f], \\ x(t_0) = x_0, \quad x(t_f) = x_T, \end{cases}$$

où l'état  $x \in AC([t_0, t_f], \mathbb{R})$  correspond à l'état de charge de la batterie et la commande  $u \in L^\infty([t_0, t_f], \mathbb{R}^2)$  au couple  $(Gear(\cdot), T_{qICE}(\cdot))$ . Les fonctions

$$\begin{aligned} f^0(t, u) &= f_{m_F}(N_W(t), 90, T_{qICE}, Gear), \\ f(t, x, u) &= f_{SOC}(T_{qW}(t), N_W(t), SOC, T_{qICE}, Gear) \end{aligned}$$

sont de classe  $\mathcal{C}^1$  en  $x$  et  $u$ , avec la notation  $x = SOC \in \mathbb{R}$  et  $u = (T_{qICE}, Gear) \in \mathbb{R}^2$ . Pour tout  $t \in [t_0, t_f]$ , l'ensemble  $U(t)$  est non vide. Enfin, l'état initial  $x_0 = SOC_0$  et l'état final  $x_T = SOC_T$  sont fixés.

**Remarque.** Le problème de commande optimale  $(P_2)$  peut être résolu en utilisant le principe du maximum non lisse de Clarke [19] ou bien le principe du maximum hybride [9]. Nous nous concentrons cependant dans ce document à la résolution du problème  $(P_3)$ .

## 1.3 État de l'art sur la modélisation

Nous venons de formuler le problème de répartition de couple comme des problèmes de commande optimale sous forme de Lagrange. Le modèle associé à ce problème est donné par un code en `Matlab Simulink` et est fourni par l'entreprise Vitesco Technologies. L'objectif est maintenant de faire un état de l'art sur les différentes formulations du problème de répartition de couple dans les véhicules hybrides électriques présentes dans la littérature. Pour cela, les trois principaux types d'architecture de véhicules hybrides sont présentés dans la section 1.3.1, et les différentes modélisations possibles sont proposées dans la section 1.3.2.

### 1.3.1 Architecture du véhicule

Il existe plusieurs types de véhicules hybrides [46] : les véhicules sans recharge (HEV pour *Hybrid Electric Vehicle*) et les véhicules avec recharge possible de la batterie (PHEV pour *Plug-in Hybrid Electric Vehicle*). Les problèmes de répartition de couple associés restent cependant très similaires. La principale différence porte sur la stratégie du choix de l'état de charge final de la batterie. Pour un HEV, on peut souhaiter conserver de l'énergie électrique pour circuler en ville, tandis que pour un PHEV on peut s'autoriser à décharger complètement la batterie si la fin du trajet correspond à une borne de recharge, ou bien au domicile du conducteur. Nous ne nous occupons pas ici de la stratégie du choix de cet état de charge final de la batterie, et considérons cette cible comme une donnée du problème.

Il existe trois principales architectures [29, 46] schématisées dans la figure 1.3 :

- **Série.** Dans l'architecture série [10, 33], la puissance mécanique générée par le moteur thermique est transformée via un générateur en puissance électrique. Cette puissance électrique est soit fournie directement au moteur électrique pour tracter le véhicule, soit stockée dans la batterie, soit les deux en même temps. L'avantage de cette architecture est que le moteur thermique peut fonctionner dans sa région d'efficacité optimale. Cependant, le coût de cette architecture est relativement élevé du fait de la présence du générateur et d'un moteur électrique puissant. De plus, en raison des conversions énergétiques mécanique/électrique, le rendement est modéré.
- **Parallèle.** Dans les véhicules hybrides parallèles [2, 44, 45, 52, 53, 61], les deux moteurs sont connectés aux roues via une transmission mécanique. Ils peuvent fonctionner simultanément ou bien séparément. Cette architecture est appréciée car elle permet un poids, un volume et un coût de construction faible. Cependant, par rapport aux véhicules à architecture série, il est plus compliqué de commander le moteur thermique dans les zones de combustion optimale.
- **Série-Parallèle.** L'architecture série-parallèle [29, 8] est composée de deux moteurs électriques : un connecté aux roues via la boîte à vitesses et l'autre au moteur thermique. Cette architecture bénéficie des avantages des deux architectures précédentes, mais elle est plus lourde, plus volumineuse et plus chère.

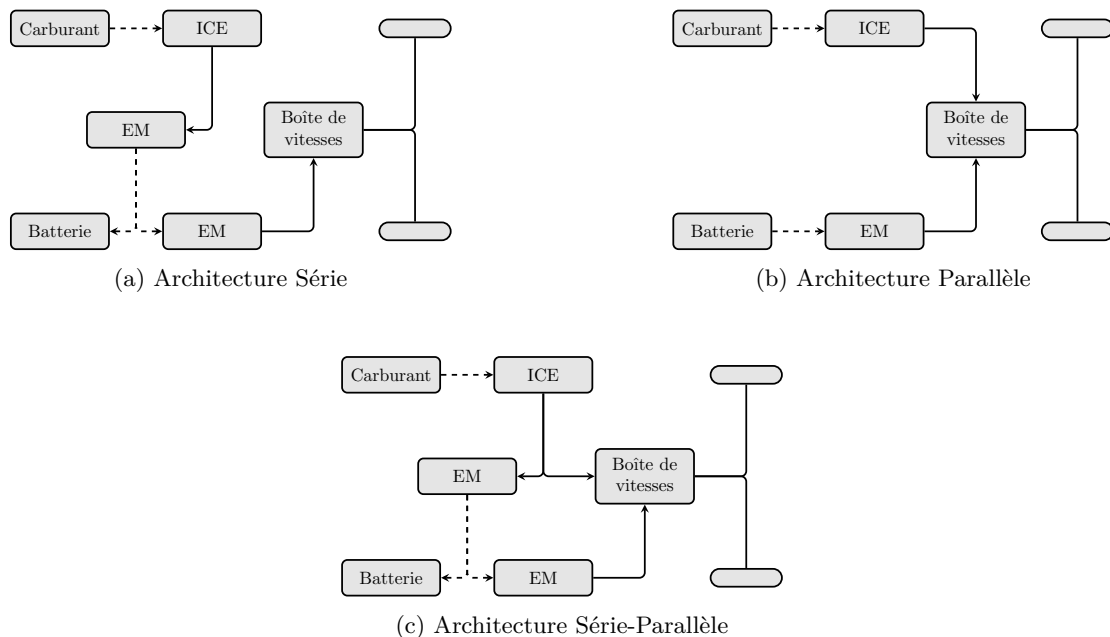


FIGURE 1.3 – Schéma des principales architectures des HEV. Les lignes pleines correspondent aux connexions mécaniques, et les lignes discontinues aux connexions énergétiques.

Le véhicule étudié a une architecture parallèle dite P3 selon la classification [33] basée sur les places respectives des moteurs électrique(s) et thermique, c'est-à-dire que les deux moteurs sont reliés à l'arbre principal au niveau de la boîte à vitesses.

**Remarque.** Il existe des formulations du problème qui prennent en compte le bilan électrique de l'alternateur (moteur électrique P0) soit par ajout d'une pénalité, soit par l'introduction d'une commande et d'un état supplémentaires [49]. Dans notre cas, nous avons constaté que ce bilan électrique est négligeable et nous avons choisi de ne pas considérer cet équipement.

### 1.3.2 Modélisation

Les dynamiques du véhicule sont des fonctions complexes. L'approche historique des constructeurs automobiles est d'utiliser des interpolations à partir de données mesurées sur des bancs moteurs [41, 44, 72], ce qui a l'avantage d'être assez précis et facilement modifiable. Cependant, les modèles ainsi obtenus sont difficilement interprétables, et le type d'interpolation a un impact conséquent sur les propriétés des dynamiques.

**États.** La modélisation la plus étudiée dans la littérature est celle à un seul état : l'état de charge de la batterie [61, 51, 41, 52, 8, 72, 2, 44, 76]. L'introduction de la température dans les états considérés a plusieurs objectifs. Le premier est d'obtenir une solution plus proche de la solution optimale du problème réel, en considérant un modèle plus proche de la réalité. En effet, la température du moteur thermique a un impact direct sur la consommation en carburant, car plus l'huile du moteur est chaude, moins il y a de frottement (et donc de pertes). Le second objectif est de faire un pas en direction de la résolution de problèmes plus complexes tels que :

- la prise en compte dans le coût d'un terme de confort basé sur l'activation d'un chauffage (lié à la température du liquide de refroidissement de la voiture).
- la prise en compte de l'émission de gaz ( $\text{NO}_x$ ,  $\text{CO}_2$ , particules fines ...). La température du moteur a en effet un impact direct sur la température des gaz à l'entrée du catalyseur [77] et on pourrait chercher à la contrôler afin de rendre le catalyseur plus efficace et ainsi réduire l'émission de gaz nocifs [35].

**Commande.** Le couple aux roues est imposé par le conducteur et la commande continue permet de choisir une répartition entre les moteurs thermique et électrique. D'après [61], les trois choix suivants de modélisation du contrôle sont semblables :

- puissance thermique [37] ou couple thermique [2, 8, 44, 61],
- puissance électrique [41] ou couple électrique,
- un ratio de répartition de puissance (PSR) [45, 76] ou de couple (TSR).

La différence de modélisation entre puissance et couple semble ne pas avoir de conséquence sur le résultat [73]. La sélection du rapport de la boîte à vitesses est également étudiée dans la littérature [53, 45, 61, 52].

D'autres commandes peuvent enfin être prises en compte, comme la puissance d'un système de chauffage électrique du pot catalytique [35].

## 1.4 État de l'art sur les méthodes de résolution

La résolution du problème de répartition de couple est bien documentée dans la littérature [25, 46, 47, 74]. L'objectif est ici de décrire les différentes méthodes utilisées.

### 1.4.1 Méthodes approchées

Les méthodes présentées dans cette section ont pour objectif d'être embarquées dans un véhicule. Elles se doivent donc d'être particulièrement rapides, peu coûteuses en place mémoire et assez robuste (contrôle en boucle fermée), éventuellement au détriment de la qualité de la solution obtenue.

**À base de règles (rule-based).** Il s'agit essentiellement d'heuristiques créées par des experts qui prennent en compte des points de fonctionnement de la cartographie moteur de la voiture. Il existe une grande diversité des méthodes à base de règles (méthode du thermostat [40], machine d'état [54], logique floue [36], ...). Rarement optimales, elles ont l'avantage d'être robustes et facilement embarquables. Un bon panorama des différentes méthodes à base de règles est réalisé dans [33, 59].

**Equivalent Consumption Minimization Strategy (ECMS).** Introduite la première fois dans [53], l'idée est d'ajouter à l'intégrande du coût de Lagrange la déviation instantanée de l'état de charge, pondérée par un coefficient  $\lambda$ . Cette formulation permet de simplifier la résolution du problème en le transformant en un problème d'optimisation pour chaque instant. Cette méthode peut être interprétée comme l'application du principe du maximum de Pontryagin [62, 75] sous certaines hypothèses assez restrictives, avec une correspondance directe entre  $\lambda$  et le co-état associé à l'état de charge de la batterie. L'ECMS a été déclinée en plusieurs versions, en fonction du choix de  $\lambda$  et de sa possible adaptation : (A-ECMS) [51], (IA-ECMS) [76], (Predictive ECMS) [39], (Map-Based ECMS) [63], (ANN-ECMS) [75], (NN-ECMS) [17].

**Model Predictive Control.** Ces méthodes décrites dans [8, 77], sans pouvoir garantir l'optimalité globale, peuvent être implémentées en ligne avec un horizon de temps assez court. De plus, des simplifications (linéarisation, approximation, auto-adaptations, ...) sur la modélisation sont souvent utilisées afin de permettre une résolution rapide et peu coûteuse.

### 1.4.2 Méthodes basées sur la théorie du contrôle optimal

Les méthodes suivantes sont basées sur la théorie du contrôle optimal. Elles sont présentées d'un point de vue algorithmique dans [13].

**Méthodes directes.** L'idée principale des méthodes directes est de transformer le problème d'optimisation en dimension infinie (le problème de commande optimale) en problème d'optimisation de dimension finie. Pour cela, la version la plus simple est de discrétiser le temps, l'état et la commande et de résoudre le problème d'optimisation associé [67] en utilisant un solveur d'optimisation classique. Ces méthodes ont l'avantage d'être faciles à mettre en œuvre et permettent une prise en compte simple des contraintes sur l'état. De plus, elles sont peu sensibles à l'initialisation du solveur d'optimisation. Cependant, ces méthodes sont assez peu précises, elles peuvent converger vers un minimum local et elles sont assez gourmandes en mémoire [70].

**Programmation dynamique.** La méthode de programmation dynamique a été introduite par Bellman [3]. Elle utilise les équations de Hamilton-Jacobi-Bellman, qui fournissent des conditions nécessaires et suffisantes d'optimalité, afin de trouver la trajectoire d'état (et les commandes associées) optimale. Cette méthode est utilisée pour résoudre le problème de commande optimale étudié ici, en version déterministe [10, 18, 45, 52, 58, 61, 72], et en version stochastique [37]. L'un des avantages de cette méthode est qu'elle fournit le contrôle en boucle fermée, ce qui est connu pour être assez robuste aux perturbations externes. Cependant, cette méthode est basée sur une exploration assez exhaustive des trajectoires possibles et elle est donc très coûteuse en termes de temps de calcul et de mémoire. Elle est également soumise au fléau de la dimension, c'est-à-dire que la complexité augmente exponentiellement en fonction de la dimension de l'état et de la commande. Une approche de programmation dynamique à deux échelles a été développée afin de tendre vers une solution embarquable [32].

**Méthodes indirectes.** Introduit dans les années 1960, le Principe du Maximum de Pontryagin (PMP) [43] permet de donner des conditions nécessaires d'optimalité d'un problème de commande optimale. Les méthodes indirectes sont basées sur ce principe, et permettent de trouver une solution du problème de commande optimale en comparant les trajectoires qui satisfont ces conditions nécessaires. La méthode indirecte classique est la méthode de tir simple. Elle consiste à chercher les zéros d'une fonction appelée fonction de tir, généralement en utilisant un solveur de type Newton. Elle a été utilisée pour résoudre notre problème [2, 29, 41]. Le contrôle est obtenu en boucle ouverte, ce qui est connu pour être moins robuste que les contrôles en boucle fermée en cas de perturbations du modèle. Ces méthodes indirectes sont réputées pour être sensibles à l'allure de la fonction de tir et à l'initialisation du solveur et elles demandent une bonne connaissance du système étudié (contraintes sur la dynamique, structure de la trajectoire optimale, ...). De plus, la prise en compte des contraintes sur l'état est complexe. Cependant, ces méthodes sont connues pour être précises, efficaces en toute dimension et moins coûteuses en mémoire que les méthodes directes ou les méthodes de programmation dynamique.

### 1.4.3 Méthodes à base d'intelligence artificielle

Plusieurs approches basées sur l'intelligence artificielle, principalement sur les réseaux de neurones, ont été utilisées dans la littérature afin de pouvoir résoudre le problème de commande optimale étudié et/ou permettre son embarquabilité.

**Apprentissage par Renforcement profond.** L'apprentissage par renforcement est basé, comme la méthode de programmation dynamique, sur les équations de Hamilton-Jacobi-Bellman et permettent d'avoir le contrôle en boucle fermée par retour d'état. Les premières méthodes d'apprentissage par renforcement ont été introduites à la fin des années 1980 dans [65]. Les méthodes d'apprentissage par renforcement profond consistent à approcher la fonction valeur par un réseau de neurones profond [66]. Ces méthodes sont utilisées pour résoudre des problèmes de commande optimale complexes et ont été appliquées à notre type de problème [35].

**Apprentissage commande-état.** L'idée principale de type de méthodes est d'avoir construit au préalable une base de données de solutions, à l'aide d'une autre méthode déjà listée. On construit ensuite un réseau de neurones qui va apprendre à fournir la commande optimale en boucle fermée en fonction des données d'entrée du véhicule (vitesse, accélération, état de charge de la batterie ...). Des exemples de cette approche ont été développés dans [18, 50].

Une approche basée sur le principe du maximum utilisant deux réseaux de neurones, l'un pour prédire l'état et l'autre pour le calcul de la commande, a été développée [68] dans le cadre de l'étude d'un bus hybride.

**Apprentissage du co-état (PMP/ECMS).** L'une des principales difficultés des méthodes PMP et ECMS est l'initialisation du co-état pour le PMP ou du facteur équivalent pour l'ECMS. Des approches basées sur l'apprentissage de ces paramètres par un réseau de neurones ont été développées [17]. Dans [75], la méthode proposée est basée sur une résolution préalable de problèmes de commande optimale par une méthode indirecte. Le réseau de neurones permet de prédire une bonne initialisation du co-état et, grâce à la relation entre le co-état et  $\lambda$ , de fournir en temps réel une bonne valeur de ce dernier.

La difficulté à justifier le bon comportement d'un réseau de neurones partout sur son domaine d'utilisation pousse les constructeurs à ne pas utiliser ce type de solution pour contrôler des parties vitales des véhicules. Cette problématique freine l'utilisation des méthodes d'apprentissage de la commande et d'apprentissage par renforcement, mais elle n'a pas d'impact sur la méthode d'apprentissage du co-état. En effet, dans cette méthode, le réseau permet une meilleure prédiction d'un paramètre, mais n'a pas d'influence directe sur la commande.

## Conclusion

Nous venons de proposer une modélisation du problème de la répartition de couple dans les véhicules hybrides sous la forme de problèmes de commande optimale. En fonction des simplifications considérées, il peut se formuler comme

- ( $P_1$ ) : un problème de commande optimale non lisse assez complexe, avec contraintes d'état,
- ( $P_2$ ) : un problème de commande optimale non lisse,
- ( $P_3$ ) : un problème de commande optimale classique.

Nous avons également donné un aperçu des méthodes utilisées dans la littérature pour résoudre ce type de problème, avec une discussion rapide des avantages et des inconvénients. Les méthodes indirectes, basées sur le principe du maximum, ont été choisies pour leur rapidité de calcul, le nombre de calculs assez faible ainsi que leur bonne précision.

## Méthodes indirectes

---

<b>2.1</b>	<b>Problème de commande optimale et conditions nécessaires .</b>	<b>22</b>
2.1.1	Problème de commande optimale . . . . .	22
2.1.2	Principe du maximum de Pontryagin . . . . .	23
2.1.3	Hypothèse principale . . . . .	26
<b>2.2</b>	<b>Méthodes de résolution indirectes . . . . .</b>	<b>30</b>
2.2.1	Méthode de tir simple . . . . .	30
2.2.2	Méthode de tir multiple . . . . .	33
<b>2.3</b>	<b>Parallélisation . . . . .</b>	<b>33</b>
2.3.1	Problème parallélisé . . . . .	34
2.3.2	Lien avec le problème initial . . . . .	35
2.3.3	Conditions nécessaires . . . . .	36
2.3.4	Lien entre les conditions nécessaires d'optimalité . . . . .	37
2.3.5	Schéma commutatif . . . . .	40
2.3.6	Méthode numérique . . . . .	40

Un problème de commande optimale (**OCP**) général et classique est introduit dans ce chapitre. L'application du principe du maximum de Pontryagin à ce type de problèmes mène à la résolution des conditions nécessaires (**NCOCP**) qui n'est généralement pas triviale.

Afin de résoudre ce problème par les méthodes dites de tir, nous introduisons l'hypothèse (**H2**), qui considère principalement que l'application exponentielle est bien définie. Bien qu'elle puisse sembler assez restrictive, nous fournissons deux cadres classiques, à savoir le cadre de Lagrange strict et le cadre hamiltonien, ainsi qu'un exemple simple (**E**), dans lesquels cette application est bien définie. Nous introduisons finalement les méthodes classiques de tir simple et multiple.

L'ensemble de ces éléments est bien connu de la littérature. Nous proposons cependant un nouveau point de vue sur la méthode de tir multiple, comme étant l'application de la méthode de tir simple à un problème parallélisé que nous appellerons (**POCP**), et dont on montrera qu'il est équivalent à (**OCP**).



## 2.1 Problème de commande optimale et conditions nécessaires

Nous introduisons le problème de commande optimale (OCP) dans la section 2.1.1, et décrivons dans la section 2.1.2 ses conditions nécessaires d'optimalité données par le principe du maximum (Théorème 2.1.2). Finalement, nous nous plaçons dans un cadre où il est plus simple de résoudre ces conditions nécessaires dans la section 2.1.3, en considérant l'hypothèse (H2).

### 2.1.1 Problème de commande optimale

On considère un système dynamique contrôlé non autonome donné par

$$\dot{x}(t) = f(t, x(t), u(t)), \quad t \in [t_0, t_f] \text{ p.p.}, \quad (2.1)$$

où le temps initial  $t_0$  et le temps final  $t_f$  sont fixés, avec  $t_0 < t_f$ . La fonction de dynamique d'état  $f: \mathbb{R} \times \mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R}^n$  est de classe  $\mathcal{C}^1$  en  $x$  et  $u$ . De plus, pour tout temps  $t \in [t_0, t_f]$ , le contrôle  $u(t)$  est à valeurs dans un fermé non vide

$$u(t) \in U(t), \quad t \in [t_0, t_f]. \quad (2.2)$$

L'état  $x$  doit satisfaire les contraintes initiales et finales

$$c(x(t_0), x(t_f)) = 0, \quad (2.3)$$

où  $c: \mathbb{R}^n \times \mathbb{R}^n \rightarrow \mathbb{R}^p$ , avec  $p \leq 2n$ , est de classe  $\mathcal{C}^1$ . De plus,  $c$  est une submersion sur  $M = c^{-1}(\{0_{\mathbb{R}^p}\})$ , c'est-à-dire que  $c'(a, b)$ , la différentielle de  $c$  au point  $(a, b)$ , est surjective pour tout  $(a, b) \in M$ . L'objectif est de trouver  $(x, u) \in \mathcal{D} := \text{AC}([t_0, t_f], \mathbb{R}^n) \times L^\infty([t_0, t_f], \mathbb{R}^m)$  qui minimise le coût de Lagrange

$$J(x, u) = \int_{t_0}^{t_f} f^0(t, x(t), u(t)) dt, \quad (2.4)$$

où  $f^0: \mathbb{R} \times \mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R}$  est une fonction de classe  $\mathcal{C}^1$  en  $x$  et  $u$ . Le problème de commande optimale décrit ci-dessus peut être formulé sous forme plus compacte de la façon suivante

$$(OCP) \quad \begin{cases} \min_{x, u} \int_{t_0}^{t_f} f^0(t, x(t), u(t)) dt \\ \text{s.c. } \dot{x}(t) = f(t, x(t), u(t)), & t \in [t_0, t_f] \text{ p.p.}, \\ u(t) \in U(t), & t \in [t_0, t_f], \\ c(x(t_0), x(t_f)) = 0. \end{cases}$$

Le problème (OCP) est un problème de commande optimale dit sous forme de Lagrange, parce que le coût (2.4) s'exprime sous forme intégrale. Un problème de commande optimale est dit sous forme de Mayer (resp. de Bolza) si le coût (2.4) est remplacé par

$$J(x, u) = g(x(t_f)) \quad \left( \text{resp. } J(x, u) = \int_{t_0}^{t_f} f^0(t, x(t), u(t)) dt + g(x(t_f)) \right).$$

Ces trois formulations sont équivalentes [16].

**Définition 2.1.1 – Point admissible et solution**

Un point admissible de (OCP) est un couple

$$(x, u) \in \mathcal{D} := \text{AC}([t_0, t_f], \mathbb{R}^n) \times L^\infty([t_0, t_f], \mathbb{R}^m)$$

qui satisfait la dynamique d'état (2.1), les contraintes sur le contrôle (2.2) et les contraintes sur les états initiaux et finaux (2.3).

Une solution de (OCP) est un point admissible  $(x, u)$  qui minimise le coût (2.4), c'est-à-dire que pour tout autre point  $(\bar{x}, \bar{u})$  admissible, on a

$$J(x, u) \leq J(\bar{x}, \bar{u}).$$

**2.1.2 Principe du maximum de Pontryagin**

Le principe du maximum de Pontryagin [43, 69] fournit des conditions nécessaires d'optimalité pour le problème (OCP).

**Théorème 2.1.2 – Principe du maximum de Pontryagin**

Si  $(x, u)$  est une solution de (OCP), alors il existe un co-état  $p \in \text{AC}([t_0, t_f], \mathbb{R}^n)$  et  $p^0 \leq 0$  tels que  $(p^0, p) \neq 0$  et tels que la dynamique hamiltonienne est satisfaite pour presque tout  $t \in [t_0, t_f]$  :

$$\begin{aligned} \dot{x}(t) &= \nabla_p h(t, x(t), p(t), p^0, u(t)), \\ \dot{p}(t) &= -\nabla_x h(t, x(t), p(t), p^0, u(t)), \end{aligned} \tag{2.5}$$

tout comme la condition de maximisation pour presque tout  $t \in [t_0, t_f]$  :

$$h(t, x(t), p(t), p^0, u(t)) = \max_{w \in U(t)} h(t, x(t), p(t), p^0, w), \tag{2.6}$$

ainsi que la condition de transversalité :

$$(-p(t_0), p(t_f)) \perp T_{(x(t_0), x(t_f))} M, \tag{2.7}$$

où  $h$  est le pseudo hamiltonien défini par

$$h(t, x, p, p^0, u) := p^0 f^0(t, x, u) + (p \mid f(t, x, u)).$$

**Remarque.** Dans le théorème précédent, comme la fonction  $c$  est une submersion sur  $M = c^{-1}(\{0_{\mathbb{R}^p}\})$ , l'espace  $M$  est une sous-variété différentielle de  $\mathbb{R}^{2n}$  de dimension  $2n - p$ . Pour tout  $v \in M$ ,  $T_v(M)$  correspond à la notation usuelle de l'espace tangent de  $M$  au point  $v$ , du point de vue des variétés différentielles. La définition canonique de cet espace est donnée dans l'annexe A. Du point de vue de l'analyse, lorsque  $M$  est un sous-ensemble de  $\mathbb{R}^{2n}$ , une « généralisation » de l'espace tangent correspond au cône tangent de Bouligand de l'ensemble  $M$  au point  $v$ , et est noté usuellement  $T_M(v)$ . On utilisera cette notation dans le chapitre 4, et on fera donc la distinction entre l'espace tangent  $T_v(M)$  à une variété différentielle  $M$  et le cône tangent  $T_M(v)$  à un ensemble  $M$ .

### Définition 2.1.3 – Extrémales

- Une *extrémale* est un quadruplet

$$(x, p, p^0, u) \in \text{AC}([t_0, t_f], \mathbb{R}^n) \times \text{AC}([t_0, t_f], \mathbb{R}^n) \times \mathbb{R}^- \times L^\infty([t_0, t_f], \mathbb{R}^m)$$

qui satisfait la dynamique hamiltonienne (2.5) ainsi que la condition de maximisation (2.6) et tel que  $(p^0, p) \neq 0$ .

- Une *BC-extrémale* (BC pour *Boundary Conditions*) est une extrémale qui satisfait les conditions aux limites sur l'état (2.3) et les conditions de transversalité (2.7).
- Une extrémale est dite *normale* si  $p^0 < 0$  et *anormale* si  $p^0 = 0$ .

Dans l'ensemble de cette thèse, on supposera implicitement l'existence d'une solution du problème (OCP), et on s'intéressera à la recherche des BC-extrémales associées.

**Dynamique hamiltonienne.** On propose de regrouper l'état  $x \in \text{AC}([t_0, t_f], \mathbb{R}^n)$  et le co-état  $p \in \text{AC}([t_0, t_f], \mathbb{R}^n)$  en une seule variable  $z = (x, p) \in \text{AC}([t_0, t_f], \mathbb{R}^{2n})$ . Le pseudo-hamiltonien peut maintenant être formulé de la manière suivante

$$h(t, z, p^0, u) = h(t, x, p, p^0, u),$$

pour tout  $t \in [t_0, t_f]$ , pour tout  $z = (x, p) \in \mathbb{R}^{2n}$  et pour tout  $p^0 \in \mathbb{R}$ . En utilisant ces notations, on remarque que les équations (2.5) sont équivalentes à

$$\dot{z}(t) = \vec{h}(t, z(t), p^0, u(t)), \quad t \in [t_0, t_f] \text{ p.p.}$$

où  $\vec{h}$  est le *champ de vecteur pseudo-hamiltonien*, défini à partir de la matrice symplectique  $\mathbf{J}_{2n}$  par

$$\vec{h}(t, z, p^0, u) = \mathbf{J}_{2n} \nabla_z h(t, z, p^0, u) \quad \text{avec} \quad \mathbf{J}_{2n} = \begin{bmatrix} \mathbf{0}_n & \mathbf{I}_n \\ -\mathbf{I}_n & \mathbf{0}_n \end{bmatrix}, \quad (2.8)$$

avec  $\mathbf{I}_n$  la matrice identité et  $\mathbf{0}_n$  la matrice nulle de  $\mathcal{M}_{n,n}(\mathbb{R})$ . Comme décrit rapidement dans l'annexe A, la matrice symplectique  $\mathbf{J}_{2n} \in \mathcal{M}_{2n,2n}(\mathbb{R})$  est à l'origine de la dynamique hamiltonienne et permet de définir canoniquement le champ de vecteur hamiltonien à partir d'une fonction réelle lisse définie sur le fibré co-tangent, appelé le hamiltonien.

Afin de pouvoir retrouver l'état  $x$  et le co-état  $p$  à partir de  $z$ , on définit la projection sur l'état  $\pi_x$  et la projection sur le co-état  $\pi_p$  pour tout  $z = (x, p)$  par

$$\pi_x(z) = x \quad \text{et} \quad \pi_p(z) = p.$$

Ces projections seront utilisées indifféremment si  $z \in \text{AC}([t_0, t_f], \mathbb{R}^{2n})$  ou si  $z \in \mathbb{R}^{2n}$ .

**Condition de transversalité.** Pour tout  $v \in M$ , l'espace tangent à  $M$  en  $v$  est donné par  $T_v M = \text{Ker}(c'(v))$ . La condition de transversalité (2.7) peut être écrite de différentes manières. D'une part, en utilisant une base  $(b_1(v), \dots, b_{2n-p}(v))$  de  $\text{Ker}(c'(v))$  et en formant la matrice

$$B_c(v) = [b_1(v) \cdots b_{2n-p}(v)] \in \mathcal{M}_{2n, 2n-p}(\mathbb{R}),$$

on a pour tout  $u \in \mathbb{R}^{2n}$

$$u \perp T_v M \iff B_c(v)^\top u = 0.$$

D'autre part, comme  $(\text{Ker } c'(v))^\perp = \text{Im}(c'(v)^\top)$ , on a pour tout  $u \in \mathbb{R}^{2n}$

$$u \perp T_v M \iff u \in \text{Im}(c'(v)^\top) \iff \exists \lambda \in \mathbb{R}^p : u = c'(v)^\top \lambda.$$

Finalement, en notant le couple état/co-état initial  $z_0 = (x_0, p_0)$  et final  $z_f = (x_f, p_f)$ , et en construisant la fonction

$$c^*(z_0, z_f) = B_c(x_0, x_f)^\top \begin{bmatrix} -p_0 \\ p_f \end{bmatrix},$$

la condition de transversalité (2.7) peut être remplacée par une des deux conditions suivantes

$$c^*(z(t_0), z(t_f)) = 0, \tag{2.9}$$

$$\exists \lambda \in \mathbb{R}^p, \quad c'(x(t_0), x(t_f))^\top \lambda = (-p(t_0), p(t_f)). \tag{2.10}$$

La formulation (2.10) est certainement une des plus utilisées pour décrire les conditions de transversalité. Cependant, on remarque que la formulation (2.9) ne contient pas  $\lambda$ , ce qui permet de réduire le nombre de variables/inconnues dans les méthodes numériques indirectes et de simplifier les notations. En effet, grâce à cette formulation, les conditions initiales et finales en état (2.3) et co-état (2.7) peuvent être réécrites de la manière suivante

$$g(z(t_0), z(t_f)) = 0,$$

où la fonction  $g: \mathbb{R}^{2n} \times \mathbb{R}^{2n} \rightarrow \mathbb{R}^{2n}$  correspondant aux contraintes initiales et finales d'état et de co-état est définie par

$$g(z_0, z_f) = (c(x_0, x_f), c^*(z_0, z_f)),$$

avec  $z_0 = (x_0, p_0)$  et  $z_f = (x_f, p_f)$ .

**Conditions nécessaires.** Avec les notations précédentes, le principe du maximum mène à la recherche des solutions de

$$\begin{cases}
 \dot{z}(t) = \vec{h}(t, z(t), p^0, u(t)), & t \in [t_0, t_f] \text{ p.p.}, & (2.11) \\
 h(t, z(t), p^0, u(t)) = \max_{w \in U(t)} h(t, z(t), p^0, w), & t \in [t_0, t_f] \text{ p.p.}, & (2.12) \\
 g(z(t_0), z(t_f)) = 0, & & (2.13) \\
 (\pi_p(z), p^0) \neq 0, & & (2.14) \\
 p^0 \leq 0. & & (2.15)
 \end{cases}$$

Avec les notations introduites, une extrémale est donc un triplet

$$(z, p^0, u) \in \text{AC}([t_0, t_f], \mathbb{R}^{2n}) \times \mathbb{R}^- \times L^\infty([t_0, t_f], \mathbb{R}^m)$$

qui satisfait les équations (2.11), (2.12), (2.14). On remarque qu'un tel triplet satisfait l'équation (2.15) par construction. Une BC-extrémale est une extrémale qui satisfait les conditions aux bords données par l'équation (2.13), c'est-à-dire une solution de (NCOCP).

### 2.1.3 Hypothèse principale

Dans le cas général, la résolution du problème (NCOCP) n'est pas simple, et principalement à cause de la condition de maximisation (2.12). Nous allons considérer une hypothèse principale de notre point de vue, qui permettra d'introduire simplement les méthodes de résolution numériques classiques du problème (NCOCP). Pour ce faire, pour deux temps intermédiaires  $t_0 \leq t_1 < t_2 \leq t_f$ , un couple état/co-état  $z_1 = (x_1, p_1) \in \mathbb{R}^n \times \mathbb{R}^n$  au temps  $t_1$  ainsi que  $p^0 \in \mathbb{R}$ , tels que  $(p^0, p_1) \neq 0$ , on définit la possible multi-fonction  $\exp_{\vec{h}} : (t_2, t_1, z_1, p^0) \mapsto z(t_2)$ , où  $z(\cdot)$  est solution du problème suivant

$$\begin{cases}
 \dot{z}(t) = \vec{h}(t, z(t), p^0, u(t)), & t \in [t_1, t_2] \text{ p.p.}, \\
 h(t, z(t), p^0, u(t)) = \max_{w \in U(t)} h(t, z(t), p^0, w), & t \in [t_1, t_2] \text{ p.p.} \\
 z(t_1) = z_1.
 \end{cases} \quad (2.16)$$

On propose de se placer dans un cadre d'étude simple, à l'aide de l'hypothèse suivante.

**Hypothèse H2** (Hypothèse principale).

La fonction  $\exp_{\vec{h}}$  est une application (ses valeurs sont des singletons), définie pour tout  $t_0 \leq t_1 < t_2 \leq t_f$ , et pour tout  $z_1 = (x_1, p_1) \in \mathbb{R}^n \times \mathbb{R}^n$  tels que  $(p_1, p^0) \neq 0$ .

**Remarque.** Le principal intérêt de cette hypothèse est de considérer que la fonction  $\exp_{\vec{h}}$  n'est pas une multi-fonction, c'est-à-dire qu'il existe au maximum une solution au temps final  $t_2$  du problème (2.16) sur tout le domaine introduit dans cette hypothèse. Par simplicité, nous avons considéré que c'est une application, et donc qu'il existe une unique solution de (2.16) sur tout ce domaine. Cependant, ce domaine de définition peut être réduit au voisinage des extrémales d'intérêt, sans introduire de difficultés supplémentaires. C'est d'ailleurs ce qui sera fait lors des expérimentations

numériques sur le problème (P<sub>3</sub>) dans le chapitre 5, où l'on remarquera avec la figure 5.2 que la fonction  $\exp_{\vec{h}}$  est une multi-application sur le domaine introduit par l'hypothèse (H2), mais est une application si on se restreint son domaine de définition au voisinage des extrémales d'intérêt.

Sous cette hypothèse, on appelle  $\exp_{\vec{h}}$  l'*application exponentielle* du champ de vecteur pseudo-hamiltonien  $\vec{h}$ . Malgré le fait que cette hypothèse semble assez restrictive, nous allons proposer deux cadres dans lesquels  $\exp_{\vec{h}}$  est une application bien définie, à savoir le cadre de Legendre strict et le cadre hamiltonien. De plus, nous allons étudier un problème de commande optimale simple, qui n'entre pas dans les deux cadres précédents, mais constater que l'hypothèse (H2) y est pourtant vérifiée.

**Cadre de Legendre strict.** Supposons que la matrice hessienne suivante est définie négative

$$\frac{\partial^2 h}{\partial u^2}(t, z, p^0, u) \prec 0,$$

pour tout  $u \in V(t) \supset U(t)$ , pour tout  $t \in [t_0, t_f]$ ,  $z \in \mathbb{R}^{2n}$  et  $p^0 \leq 0$ , où  $V(t)$  est un ensemble ouvert de  $\mathbb{R}^m$ . Sous cette hypothèse, la condition de Legendre stricte est vérifiée le long de toutes les extrémales  $(z, p^0, u)$ , voir [1, Prop. 20.11]. Pour tout temps  $t \in [t_0, t_f]$ , le contrôle  $u(t)$  satisfait les conditions nécessaires et suffisantes d'optimalité à l'ordre 2. Il est donc possible de remplacer la condition (2.12) par

$$\nabla_u h(t, z(t), p^0, u(t)) = 0, \quad t \in [t_0, t_f] \text{ p.p.}$$

De plus, en utilisant le théorème des fonctions implicites, il existe une fonction  $\bar{u}(t, z, p^0)$  qui nous fournit la commande sous forme fermée (ou *feedback*). Le problème (2.16) peut donc se réécrire sous la forme suivante

$$\begin{cases} \dot{z}(t) = \vec{h}(t, z(t), p^0, \bar{u}(t, z(t), p^0)), & \forall t \in [t_1, t_2] \text{ p.p.}, \\ z(t_1) = z_1. \end{cases}$$

En utilisant le théorème d'existence et d'unicité des solutions maximales, on peut en déduire que l'application exponentielle  $\exp_{\vec{h}}(t_2, t_1, z_1, p^0)$  est bien définie. De plus, si on suppose que son domaine de définition correspond à celui introduit dans l'hypothèse (H2), cette hypothèse est donc bien vérifiée dans ce cadre.

**Cadre hamiltonien.** Le cadre hamiltonien est basé sur l'hypothèse suivante

**Hypothèse H3** (Hamiltonien lisse).

Pour toute extrémale  $(\bar{z}, p^0, \bar{u})$  de (OCP), pour presque tout  $t \in [t_0, t_f]$  et pour tout  $z$  dans un voisinage de  $\bar{z}(t)$ , le pseudo-hamiltonien maximisé définit un « véritable » hamiltonien  $H$  de classe  $\mathcal{C}^1$

$$H(t, z, p^0) = \max_{w \in U(t)} h(t, z, p^0, w).$$

Cette hypothèse nous permet de transformer le problème donné par le système (2.16) en un problème de Cauchy, via la proposition suivante.

**Proposition 2.1.4 – Caractérisation des extrémales [1, Prop. 12.1]**

Sous (H3), au voisinage d'une extrémale de référence, le triplet  $(z, p^0, u)$  est une extrémale de (OCP) si et seulement si

$$\begin{cases} \dot{z}(t) = \vec{H}(t, z(t), p^0), & \forall t \in [t_0, t_f] \text{ p.p.}, \\ (\pi_p(z), p^0) \neq 0, \end{cases} \quad (2.17)$$

où  $\vec{H}$  est le champ de vecteur hamiltonien défini par

$$\vec{H}(t, z, p^0) = \mathbf{J}_{2n} \nabla_z H(t, z, p^0), \quad (2.18)$$

avec

$$H(t, z, p^0) = \max_{w \in \mathbf{U}(t)} h(t, z, p^0, w).$$

► Soit  $(\bar{z}, p^0, \bar{u})$  une extrémale du problème (OCP) et  $t \in [t_0, t_f]$ . On définit

$$\begin{aligned} F(z) &= H(t, z, p^0) - h(t, z, p^0, \bar{u}(t)) \\ &= \max_{w \in \mathbf{U}(t)} h(t, z, p^0, w) - h(t, z, p^0, \bar{u}(t)). \end{aligned}$$

Sous l'hypothèse (H3), le gradient  $\nabla F$  existe et est défini sur un voisinage de  $\bar{z}(t)$ . Pour tout  $z$  dans ce voisinage,  $F(z) \geq 0$  et  $F(\bar{z}(t)) = 0$  car  $\bar{u}$  satisfait la condition de maximisation donnée par l'équation (2.12). Le point  $\bar{z}(t)$  est donc un minimum de  $F$ , et donc  $\nabla F(\bar{z}(t)) = 0$ . Autrement dit

$$\begin{aligned} \nabla_z h(t, \bar{z}(t), p^0, \bar{u}(t)) &= \nabla_z H(t, \bar{z}(t), p^0) \\ \implies \dot{\bar{z}}(t) &= \vec{H}(t, \bar{z}(t), p^0). \end{aligned}$$

Réciproquement, s'il existe  $(\bar{z}, p^0) \in \text{AC}([t_0, t_f], \mathbb{R}^{2n}) \times \mathbb{R}$  tel que  $(\pi_p(\bar{z}), p^0) \neq 0$  et tel que pour presque tout  $t \in [t_0, t_f]$ ,  $\dot{\bar{z}}(t) = \vec{H}(t, \bar{z}(t), p^0)$ , alors, par définition de  $H$ , il existe un contrôle mesurable<sup>1</sup>  $\bar{u} \in L^\infty([t_0, t_f], \mathbb{R}^m)$  qui maximise le pseudo-hamiltonien pour presque tout  $t \in [t_0, t_f]$  le long de  $\bar{z}$

$$H(t, \bar{z}(t), p^0) = \max_{w \in \mathbf{U}(t)} h(t, \bar{z}(t), p^0, w) = h(t, \bar{z}(t), p^0, \bar{u}(t)),$$

et de la même manière que précédemment, on a

$$\begin{aligned} \nabla_z H(t, \bar{z}(t), p^0) &= \nabla_z h(t, \bar{z}(t), p^0, \bar{u}(t)) \\ \implies \dot{\bar{z}}(t) &= \vec{h}(t, \bar{z}(t), p^0, \bar{u}(t)). \end{aligned}$$

Ainsi  $(\bar{z}, p^0, \bar{u})$  est bien une extrémale du problème (OCP). ■

1. Voir [1, Proposition 12.1] pour l'aspect mesurable.

Sous l'hypothèse (H3), en utilisant la proposition 2.1.4, la résolution du problème (2.16) est équivalente à la résolution du problème de Cauchy suivant

$$\begin{cases} \dot{z}(t) = \vec{H}(t, z(t), p^0), & \forall t \in [t_1, t_2] \text{ p.p.}, \\ z(t_1) = z_1. \end{cases}$$

En utilisant le théorème d'existence et d'unicité des solutions maximales, l'application exponentielle du champ hamiltonien est bien définie. De même que dans le cadre de Legendre strict, si on considère que son domaine de définition est celui introduit dans l'hypothèse (H2), alors cette hypothèse est vérifiée.

**Exemple.** On s'intéresse à l'exemple suivant

$$(E) \quad \begin{cases} \min_{x,u} \int_0^{t_f} x(t) dt \\ \text{s.c. } \dot{x}(t) = u(t), & t \in [0, t_f] \text{ p.p.}, \\ u(t) \in [-1, 1], & t \in [0, t_f], \\ x(t_0) = 0, \quad x(t_f) = x_T, \end{cases}$$

où  $t_f$  et  $x_T$  sont fixés. On remarque que ce problème appartient à la classe des problèmes étudiés dans ce chapitre, à savoir (OCP). On peut donc appliquer le principe du maximum sur (E). Le pseudo-hamiltonien est donné par

$$h(z, p^0, u) = p^0 x + p u.$$

La condition de maximisation nous fournit le contrôle sous forme fermée

$$u^*(z) = \text{signe}(p) \implies u(t) = \text{signe}(p(t)) \quad \forall t \in [0, t_f],$$

et la dynamique hamiltonienne nous fournit la dynamique du co-état

$$\dot{p}(t) = -p^0 \implies p(t) = p(0) - p^0 t, \quad \forall t \in [0, t_f].$$

La valeur de l'état au temps  $t \in [0, t_f]$  dépend donc de  $p^0$  et de la valeur du co-état initial  $p_0 = p(0)$ . Plus précisément, en notant  $t_a = \frac{p_0}{p^0}$  (lorsque  $p^0 \neq 0$ ), la trajectoire d'état est donnée par

$$x(t, p_0, p^0) = \begin{cases} t \text{ signe}(p_0) & \text{si } p^0 = 0, \\ t \text{ signe}(p^0) & \text{si } p^0 \neq 0 \text{ et } t_a > t, \\ (2t_a - t) \text{ signe}(p^0) & \text{si } p^0 \neq 0 \text{ et } 0 \leq t_a \leq t, \\ -t \text{ signe}(p^0) & \text{si } p^0 \neq 0 \text{ et } t_a < 0, \end{cases} \quad (2.19)$$

et la trajectoire du coût est donnée par

$$x^0(t, p_0, p^0) = \begin{cases} \frac{1}{2} t^2 \text{ signe}(p_0) & \text{si } p^0 = 0, \\ \frac{1}{2} t^2 \text{ signe}(p^0) & \text{si } p^0 \neq 0 \text{ et } t_a > t, \\ (-\frac{1}{2} t^2 + 2t_a t - t_a^2) \text{ signe}(p^0) & \text{si } p^0 \neq 0 \text{ et } 0 \leq t_a \leq t, \\ -\frac{1}{2} t^2 \text{ signe}(p^0) & \text{si } p^0 \neq 0 \text{ et } t_a < 0. \end{cases} \quad (2.20)$$



L'application exponentielle  $\exp_{\vec{h}}$  associée au problème (E) est donc donnée par

$$\exp_{\vec{h}}(t, p_0, p^0) = (x(t, p_0, p^0), p_0 - p^0 t)$$

pour tout  $t \in [0, t_f]$  et pour tout  $(p_0, p^0)$  non trivial.

**Remarque.** Dans cet exemple illustratif, puisque le temps initial et l'état initial de l'intégration sont tous les deux fixés à zéro, nous ne les faisons pas apparaître dans les arguments de l'application exponentielle. On pourrait cependant les faire apparaître et constater que la fonction exponentielle est bien définie sur tout le domaine introduit dans l'hypothèse (H2). Nous avons choisi simplement de définir cette application au voisinage des extrémales d'intérêts de ce problème.

L'hypothèse (H2) est donc vérifiée. Le problème (E) n'entre ni dans le cadre de Legendre strict car l'ensemble  $U = [-1, 1]$  est un fermé, ni dans le cadre hamiltonien. En effet, le vrai hamiltonien du problème (E) est donnée par

$$H(z, p^0) = p^0 x + |p|,$$

qui n'est pas une fonction de classe  $\mathcal{C}^1$ . Cependant, on remarque que pour tout  $p \neq 0$ ,

$$\vec{h}(z, u^*(z), p^0) = (u^*(z), -p^0) = (\text{signe}(p), -p^0) = \vec{H}(z, p^0).$$

**Conclusion.** Finalement, l'hypothèse (H2) n'est pas une hypothèse si restrictive. En effet, nous venons de proposer deux cadres classiques, à savoir le cadre de Legendre strict et le cadre hamiltonien, où l'application exponentielle est bien définie. De plus, l'étude du problème (E) montre qu'il existe des problèmes n'entrant dans aucun des deux cadres précédents, mais où l'hypothèse (H2) est vérifiée. À partir de maintenant, nous considérons donc que le problème (OCP) vérifie l'hypothèse (H2).

## 2.2 Méthodes de résolution indirectes

Dans cette section sont proposées des méthodes de résolution du problème de commande optimale (OCP) de type indirectes, c'est-à-dire basées sur le principe du maximum. On s'intéresse donc à la résolution du problème (NCOCP), sous l'hypothèse (H2).

### 2.2.1 Méthode de tir simple

Remarquons que si  $p^0 = 0$  alors  $\dot{p}(t)$  est linéaire en  $p(t)$

$$\dot{p}(t) = \left( p(t) \left| \frac{\partial f}{\partial x}(t, x(t), u(t)) \right. \right), \quad t \in [t_0, t_f].$$

La condition  $(p^0, p) \neq 0$  est donc équivalente à dire qu'il existe  $t \in [t_0, t_f]$  tels que  $(p^0, p(t)) \neq 0$ . Sous l'hypothèse (H2), trouver une solution de (NCOCP) revient donc à trouver  $p^0 \leq 0$ , un couple état/co-état initial  $z_0 \in \mathbb{R}^{2n}$  ainsi qu'un couple état/co-état

final  $z_f \in \mathbb{R}^{2n}$  solution de

$$(TPBVP) \quad \begin{cases} z_f = \exp_{\vec{h}}(t_f, t_0, z_0, p^0), \\ g(z_0, z_f) = 0, \\ p^0 \leq 0, \quad (p^0, \pi_p(z_0)) \neq 0. \end{cases}$$

Les deux premières équations peuvent être réécrites sous la forme compacte suivante

$$g(z_0, \exp_{\vec{h}}(t_f, t_0, z_0, p^0)) = 0. \quad (2.21)$$

**Remarque.** Un problème aux deux bouts (*Two-Points Boundary Value Problem*) est généralement un problème composé d'équations différentielles ordinaires (EDO) et de conditions aux bords. Dans notre cas, l'application exponentielle ne correspond pas nécessairement à l'intégration d'une EDO à partir d'une condition initiale fixée du fait de la condition de maximisation. Afin de garder des notations simples, nous proposons malgré tout de conserver l'appellation problème aux deux bouts pour (TPBVP).

La méthode de tir simple consiste à chercher numériquement une racine  $(z_0, p^0)$  non triviale (telle que  $(\pi_p(z_0), p^0) \neq 0$ ) de la *fonction de tir simple*  $S: \mathbb{R}^{2n} \times \mathbb{R}^- \rightarrow \mathbb{R}^{2n}$  définie par

$$S(z_0, p^0) = g(z_0, \exp_{\vec{h}}(t_f, t_0, z_0, p^0)).$$

On remarque cependant que l'on recherche  $2n + 1$  inconnues qui satisfont  $2n$  équations. En réalité, les BC-extrémales sont homogènes en  $(p^0, p)$ . En effet, si  $(x, p, p^0, u)$  est une BC-extrémale, alors pour tout  $k > 0$ , le quadruplet  $(x, kp, kp^0, u)$  est également une BC-extrémale. Pour lever cette homogénéité, on propose deux méthodes réduisant d'une unité le nombre d'inconnues du problème de recherche de zéro et qui mènent à deux fonctions de tir différentes. La première est basée sur l'hypothèse suivante

**Hypothèse H4** (Normalité des BC-extrémales).

Les BC-extrémales associées aux solutions de (OCP) sont normales.

Sous l'hypothèse (H4) et en utilisant l'homogénéité des BC-extrémales en  $(p^0, p)$ , on peut fixer  $p^0 = -1$ . La fonction de tir simple associée est donc  $S_1: \mathbb{R}^{2n} \rightarrow \mathbb{R}^{2n}$  définie par

$$S_1(z_0) = S(z_0, -1). \quad (2.22)$$

La seconde méthode exploite uniquement l'homogénéité des BC-extrémales en fixant  $\|(p^0, p)\|_2 = 1$ . Une formulation possible de la fonction de tir associée est la fonction  $S_2: \mathbb{R}^n \times \mathcal{B}^n \rightarrow \mathbb{R}^{2n}$  définie par

$$S_2(x_0, p_0) = S(x_0, p_0, \eta(p_0)), \quad (2.23)$$

avec  $\eta: \mathcal{B}^n \rightarrow \mathbb{R}^-$  la fonction définie par

$$\eta(p_0) = -\sqrt{1 - \|p_0\|_2^2}, \quad \text{et} \quad \mathcal{B}^n = \{x \in \mathbb{R}^n \mid \|x\|_2 \leq 1\}.$$

On retrouve directement par construction que pour tout  $p \in \mathcal{B}^n$ ,

$$\|(p_0, \eta(p_0))\|_2 = \sqrt{\|p_0\|_2^2 + 1 - \|p_0\|_2^2} = 1.$$

**Application sur le problème (E).** Comme dans le problème (E) l'état initial est fixé, on peut considérer directement la fonction de tir  $S: \mathbb{R} \times \mathbb{R}^- \rightarrow \mathbb{R}$  définie par

$$S(p_0, p^0) = x(t_f, p_0, p^0) - x_T$$

$$= \begin{cases} -t_f - x_T & \text{si } p_0 < p^0 t_f, \\ t_f - 2\frac{p_0}{p^0} - x_T & \text{si } p^0 \neq 0 \text{ et } p^0 t_f \leq p_0 \leq 0, \\ t_f - x_T & \text{si } p_0 > 0. \end{cases}$$

Cette fonction de tir est affichée dans la figure 2.1. On remarque que cette fonction n'est pas définie en  $(0, 0)$ , ce qui fait directement référence à la condition (2.14). Sous l'hypothèse (H4), la fonction de tir  $S_1: \mathbb{R} \rightarrow \mathbb{R}$  est donc donnée par

$$S_1(p_0) = \begin{cases} -t_f - x_T & \text{si } p_0 < -t_f, \\ t_f + 2p_0 - x_T & \text{si } -t_f \leq p_0 \leq 0, \\ t_f - x_T & \text{si } p_0 > 0. \end{cases} \quad (2.24)$$

Cette fonction de tir est linéaire par morceaux, ce qui n'est pas le cas de la fonction de tir  $S_2: [-1, 1] \rightarrow \mathbb{R}$  qui n'a pas besoin de l'hypothèse (H4) et qui est définie par

$$S_2(p_0) = \begin{cases} -t_f - x_T & \text{si } p_0 < -t_f \sqrt{1 - p_0^2}, \\ t_f + 2\frac{p_0}{\sqrt{1 - p_0^2}} - x_T & \text{si } |p_0| \neq 1 \text{ et } -\sqrt{1 - p_0^2} t_f \leq p_0 \leq 0 \\ t_f - x_T & \text{si } p_0 > 0. \end{cases}$$

Le zéro de la fonction de tir  $S_1$  est donné simplement par  $p_0 = \frac{1}{2}(x_T - t_f)$  pour tout état final admissible, c'est-à-dire pour tout  $x_T \in [-t_f, t_f]$ .

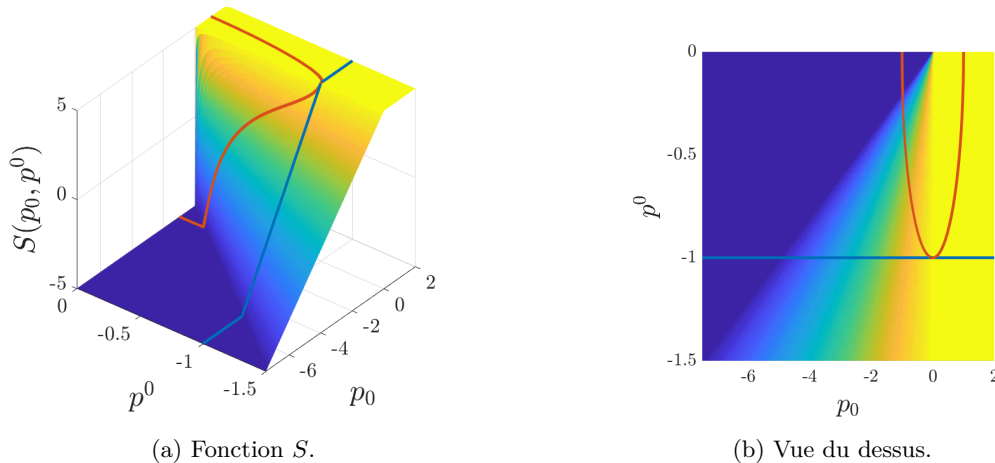


FIGURE 2.1 – Fonction de tir  $S$ , avec  $x_T = 0$  et  $t_f = 5$ . Les courbes bleu et rouge correspondent respectivement aux deux méthodes de normalisation utilisées, à savoir la restriction de  $S$  au plan  $p^0 = -1$  et à  $\|(p^0, p)\| = 1$ .

### 2.2.2 Méthode de tir multiple

La méthode de tir simple peut conduire à certains problèmes numériques, la fonction de tir simple étant connue pour être très sensible aux conditions initiales [5, 55, 56], surtout dans le cas d'un intervalle de temps long et/ou de fortes non linéarités. La méthode de tir multiple [4] a été développée afin de réduire ces problèmes numériques, en réduisant la plage d'intégration.

L'idée principale de cette méthode est d'intégrer la dynamique hamiltonienne sur des intervalles de temps plus courts. Pour cela, on décompose l'intervalle de temps  $[t_0, t_f]$  en  $N + 1$  sous-intervalles de temps  $\Delta_i = [t_i, t_{i+1}]$  pour tout  $i \in \mathbb{N}_N := \llbracket 0, N \rrbracket$ . Le problème aux deux bouts (TPBVP) est transformé en un problème aux deux bouts et aux points intermédiaires (*Multi-Points Boundary Value Problem*) suivant

$$(MPBVP) \quad \begin{cases} \exp_{\vec{h}}(t_{i+1}, t_i, z_i, p^0) = z_{i+1}, & i \in \mathbb{N}_N, \\ g(z_0, z_{N+1}) = 0, \\ p^0 \leq 0, \quad (\pi_p(z_i), p^0) \neq 0, & i \in \mathbb{N}_N. \end{cases}$$

À nouveau, ce problème peut être résolu par la méthode de tir en cherchant un zéro de la *fonction de tir multiple*  $S_m: (\mathbb{R}^{2n})^{N+1} \times \mathbb{R}^- \rightarrow (\mathbb{R}^{2n})^{N+1}$  définie par

$$S_m(z_0, \dots, z_N, p^0) = \begin{pmatrix} \exp_{\vec{h}}(t_1, t_0, z_0, p^0) - z_1 \\ \vdots \\ \exp_{\vec{h}}(t_N, t_{N-1}, z_{N-1}, p^0) - z_N \\ g(z_0, \exp_{\vec{h}}(t_{N+1}, t_N, z_N, p^0)) \end{pmatrix}, \quad (2.25)$$

tel que  $(\pi_p(z_i), p^0) \neq 0$  pour tout  $i \in \mathbb{N}_N$ . De la même manière, il est possible de réduire le nombre d'inconnues en utilisant les deux méthodes présentées pour la méthode de tir simple. Finalement, la fonction de tir multiple est connue pour être moins sensible aux conditions initiales que la fonction de tir simple [4, 64], due à l'appel de l'application exponentielle sur des intervalles de temps plus courts, ce qui rend cette méthode plus robuste. En effet, on peut montrer facilement que la divergence du pseudo-hamiltonien  $h$  est constante, ce qui implique que les trajectoires d'état et de co-état au voisinage d'une BC-extrémale vont diverger avec le même taux. L'erreur effectuée sur l'initialisation va être amplifiée tout au long de l'intégration de la dynamique pseudo-hamiltonienne, ce qui implique directement que réduire l'intervalle de temps d'intégration permet de réduire la sensibilité de la fonction de tir [5].

## 2.3 Parallélisation

Dans la méthode de tir multiple, on recherche les extrémales en dualisant le problème (OCP) à l'aide du principe du maximum pour obtenir le problème (TPBVP), puis on découpe ce problème aux deux bouts pour obtenir le problème (MPBVP), que l'on résout avec la méthode de tir. L'objectif de cette section est d'introduire une nouvelle manière d'aller de (OCP) à (MPBVP), c'est-à-dire un autre point de vue sur la méthode de tir multiple. Dans ce nouveau point de vue, on parallélise le problème (OCP) pour

obtenir le problème (POCP) équivalent, puis on dualise ce problème parallélisé pour obtenir une nouvelle expression des conditions d'optimalité (TPBVPP), qui sont une autre formulation de (MPBVP).

Pour présenter ce nouveau point de vue sur la méthode de tir multiple, nous introduisons tout d'abord dans la section 2.3.1 le problème (POCP), et nous montrons dans la section 2.3.2 que ce problème est équivalent dans un certain sens au problème (OCP). Nous formulons dans la section 2.3.3 les conditions nécessaires d'optimalité (NCPOCP) du problème parallélisé, et montrons dans la section 2.3.4 qu'elles sont équivalentes, dans un sens que nous préciserons, aux conditions nécessaires d'optimalité (NCOCP) du problème d'origine. Toute cette démarche est synthétisée au travers d'un schéma commutatif en section 2.3.5. Finalement, nous montrons dans la section 2.3.6 que la méthode de tir multiple introduite dans la section 2.2.2 peut être vue comme la méthode de tir simple appliquée au problème parallélisé (POCP).

### 2.3.1 Problème parallélisé

L'idée principale du problème de commande optimale parallèle est de paralléliser l'état et la commande sur les intervalles de temps  $\Delta_i$ . Pour cela, on définit pour tout entier  $i \in \mathbb{N}_N := \{0, \dots, N\}$  le changement de variable temporelle inversible  $\phi_i: \mathbb{R} \rightarrow \mathbb{R}$  par

$$\phi_i(s) = s t_{i+1} + (1-s)t_i.$$

Cette fonction, dont l'objectif est de parcourir les intervalles de temps  $\Delta_i$  simultanément en prenant  $s \in [0, 1]$ , permet de définir la dynamique de coût du problème parallèle  $f_p^0: \mathbb{R} \times (\mathbb{R}^n)^{N+1} \times (\mathbb{R}^m)^{N+1} \rightarrow \mathbb{R}$  par

$$f_p^0(s, \bar{x}, \bar{u}) = \sum_{i=0}^N l_i f^0(\phi_i(s), \bar{x}_i, \bar{u}_i)$$

et la dynamique d'état parallèle  $f_p: \mathbb{R} \times (\mathbb{R}^n)^{N+1} \times (\mathbb{R}^m)^{N+1} \rightarrow (\mathbb{R}^n)^{N+1}$  par

$$f_p(s, \bar{x}, \bar{u}) = \left( l_i f(\phi_i(s), \bar{x}_i, \bar{u}_i) \right)_{i \in \mathbb{N}_N}$$

où pour tout  $i \in \mathbb{N}_N$ ,  $l_i = t_{i+1} - t_i$  correspond à la longueur de l'intervalle  $\Delta_i$  (et à la dérivée de  $\phi_i$ ), et où

$$\bar{x} = (\bar{x}_0, \dots, \bar{x}_N) \in (\mathbb{R}^n)^{N+1} \quad \text{et} \quad \bar{u} = (\bar{u}_0, \dots, \bar{u}_N) \in (\mathbb{R}^m)^{N+1}.$$

Le domaine de contrôle admissible est également parallélisé

$$U_p(s) = \left( (U \circ \phi_i)(s) \right)_{i \in \mathbb{N}_N}.$$

Les contraintes initiales et finales sur l'état sont prises en compte dans la fonction de contrainte parallélisée  $c_p: (\mathbb{R}^n)^{N+1} \times (\mathbb{R}^n)^{N+1} \rightarrow \mathbb{R}^{nN+p}$  définie par

$$c_p(a, b) = (d(h), c(a_0, b_N),) \tag{2.26}$$

où  $h = (a_1, \dots, a_N, b_0, \dots, b_{N-1}) \in (\mathbb{R}^n)^{2N}$ , avec  $a = (a_0, \dots, a_N)$  et  $b = (b_0, \dots, b_N)$ . La fonction  $d: (\mathbb{R}^n)^{2N} \rightarrow (\mathbb{R}^n)^N$  est définie par  $d(h) = Lh$  avec  $L = [\mathbf{I}_{nN} \quad -\mathbf{I}_{nN}]$ , pour tout  $h \in (\mathbb{R}^n)^{2N}$ . Cette fonction linéaire  $d$  correspond simplement aux contraintes de raccordement de l'état aux extrémités des intervalles. Finalement, le problème de commande optimale parallèle est défini par

$$(POCP) \quad \begin{cases} \min_{\bar{x}, \bar{u}} \int_0^1 f_p^0(s, \bar{x}(s), \bar{u}(s)) \, ds \\ \text{s.c. } \dot{\bar{x}}(s) = f_p(s, \bar{x}(s), \bar{u}(s)), & s \in [0, 1] \text{ p.p.}, \\ \bar{u}(s) \in U_p(s), & s \in [0, 1], \\ c_p(\bar{x}(0), \bar{x}(1)) = 0. \end{cases}$$

On remarque que le problème (POCP) est un problème de commande optimale sous la même forme que (OCP), et soumis aux mêmes hypothèses, c'est-à-dire que les fonctions  $f_p^0$  et  $f_p$  sont de classe  $\mathcal{C}^1$  en  $x$  et  $u$ , la fonction  $c_p$ , également de classe  $\mathcal{C}^1$ , est une submersion sur  $M_p = c_p^{-1}(\{0_{\mathbb{R}^{nN+p}}\})$ , et pour tout  $s \in [0, 1]$ , l'ensemble  $U_p(s)$  est un fermé non vide. De plus, considérer l'hypothèse (H2) sur (OCP) revient à considérer la même hypothèse sur (POCP).

### 2.3.2 Lien avec le problème initial

L'objectif est maintenant de montrer le lien entre le problème initial (OCP) et le problème parallélisé (POCP). Pour cela, on définit le coût du problème parallélisé  $J_p: \mathcal{D}_p \rightarrow \mathbb{R}$  où  $\mathcal{D}_p = \text{AC}([0, 1], (\mathbb{R}^n)^{N+1}) \times L^\infty([0, 1], (\mathbb{R}^m)^{N+1})$ , par

$$J_p(\bar{x}, \bar{u}) = \int_0^1 f_p^0(s, \bar{x}(s), \bar{u}(s)) \, ds$$

et la fonction  $\theta_p: \mathcal{D} \rightarrow \mathcal{D}_p$  définie par

$$\theta_p(x, u) = (\mathbf{S}_{\mathbf{p},n}(x), \mathbf{S}_{\mathbf{p},m}(u))$$

où pour tout  $k > 0$ , l'opérateur de découpe  $\mathbf{S}_{\mathbf{p},k}: L^\infty(\mathbb{R}, \mathbb{R}^k) \rightarrow L^\infty(\mathbb{R}, (\mathbb{R}^k)^{N+1})$  est défini par

$$\mathbf{S}_{\mathbf{p},k}(v) = (v \circ \phi_i)_{i \in \mathbb{N}_N}$$

L'objectif est de montrer la proposition 2.3.3, qui établit l'équivalence des solutions entre les problèmes (OCP) et (POCP). Pour cela, nous allons utiliser les deux lemmes suivants, dont les preuves peuvent être trouvées dans l'annexe B.

#### Lemme 2.3.1 – Bijection sur les points admissibles

*Le point  $(x, u)$  est admissible pour (OCP) si et seulement si le point  $\theta_p(x, u)$  est admissible pour (POCP).*

► Preuve dans l'annexe B. ■

**Lemme 2.3.2 – Égalité des coûts**

Pour tout  $(x, u) \in \mathcal{D}$ , on a  $J(x, u) = J_p(\theta_p(x, u))$ .

► Preuve dans l'annexe B. ■

**Proposition 2.3.3 – Équivalence des solutions**

Le point  $(x, u)$  est solution de (OCP) si et seulement si le point  $\theta_p(x, u)$  est solution de (POCP). On note cette relation par (OCP)  $\iff$  (POCP).

► Soit  $(x^*, u^*)$  solution de (OCP). Par définition, le point  $(x^*, u^*)$  est admissible de (OCP) et pour tout autre point admissible  $(x, u)$  de (OCP), on a

$$J(x^*, u^*) \leq J(x, u).$$

On pose  $(\bar{x}^*, \bar{u}^*) = \theta_p(x^*, u^*)$ . D'après le lemme 2.3.1, le point  $(\bar{x}^*, \bar{u}^*)$  est un point admissible de (POCP).

De plus, le point  $(\bar{x}^*, \bar{u}^*)$  minimise le coût parallèle. En effet, pour tout autre point admissible  $(\bar{x}, \bar{u})$  du problème (POCP), en utilisant le lemme 2.3.1, il existe un point admissible  $(x, u) \in \mathcal{D}$  du problème (OCP) tel que  $(\bar{x}, \bar{u}) = \theta_p(x, u)$ . En utilisant le lemme 2.3.2, on a directement

$$J_p(\bar{x}^*, \bar{u}^*) = J_p(\theta_p(x^*, u^*)) = J(x^*, u^*) \leq J(x, u) = J_p(\theta_p(x, u)) = J_p(\bar{x}, \bar{u}).$$

On procède de manière analogue pour montrer la réciproque. ■

**2.3.3 Conditions nécessaires**

La proposition 2.3.3 établit l'équivalence des problèmes (OCP) et (POCP). Nous allons maintenant montrer l'équivalence entre les conditions nécessaires d'optimalité de ces deux problèmes. Comme remarqué à la fin de la section 2.3.1, le problème parallélisé (POCP) est un problème de commande optimale, sous la même forme et les mêmes hypothèses que le problème (OCP). Nous pouvons donc appliquer le principe du maximum à (POCP).

Le champ de vecteur pseudo-hamiltonien du problème parallélisé  $\vec{h}_p$  est défini canoniquement par

$$\vec{h}_p(s, \bar{z}, p^0, \bar{u}) = \mathbf{J}_{2n(N+1)} \nabla_{\bar{z}} h_p(s, \bar{z}, p^0, \bar{u})$$

à partir de  $h_p: \mathbb{R} \times (\mathbb{R}^{2n})^{N+1} \times \mathbb{R} \times (\mathbb{R}^m)^{N+1} \rightarrow \mathbb{R}$ , le pseudo-hamiltonien du problème parallélisé, défini par

$$h_p(s, \bar{z}, p^0, \bar{u}) = p^0 f_p^0(s, \bar{x}, \bar{u}) + (\bar{p} \mid f_p(s, \bar{x}, \bar{u})),$$

avec la notation  $\bar{z} = (\bar{x}, \bar{p}) \in (\mathbb{R}^n)^{N+1} \times (\mathbb{R}^n)^{N+1}$ . De même, on construit la fonction de contraintes initiales et finales d'état/co-état du problème parallélisé

$$g_p: (\mathbb{R}^{2n})^{N+1} \times (\mathbb{R}^{2n})^{N+1} \rightarrow (\mathbb{R}^{2n})^{N+1}$$

de la même manière que la fonction  $g$  pour le problème (OCP). À cet effet, pour tout élément  $v \in M_p = c_p^{-1}(\{0_{\mathbb{R}^{p+nN}}\})$  on construit la matrice  $B_{c_p}(v)$  dont les colonnes constituent une base de  $\text{Ker}(c'_p(v))$ . On a donc pour tout  $\bar{z}_0 = (\bar{x}_0, \bar{p}_0)$  et  $\bar{z}_f = (\bar{x}_f, \bar{p}_f)$

$$g_p(\bar{z}_0, \bar{z}_f) = (c_p^*(\bar{z}_0, \bar{z}_f), c_p(\bar{x}_0, \bar{x}_f))$$

où la fonction  $c_p^*: (\mathbb{R}^{2n})^{N+1} \times (\mathbb{R}^{2n})^{N+1} \rightarrow \mathbb{R}^{nN+(2n-p)}$  est définie par

$$c_p^*(\bar{z}_0, \bar{z}_f) = B_{c_p}(\bar{x}_0, \bar{x}_f)^\top \begin{bmatrix} -\bar{p}_0 \\ \bar{p}_f \end{bmatrix}.$$

Les conditions nécessaires d'optimalité du problème (POCP) données par le principe du maximum conduisent à chercher les solutions  $(\bar{z}, p^0, \bar{u})$  du problème suivant

$$\begin{cases} \dot{\bar{z}}(s) = \vec{h}_p(s, \bar{z}(s), p^0, \bar{u}(s)), & s \in [0, 1] \text{ p.p.}, & (2.27) \\ h_p(s, \bar{z}(s), p^0, \bar{u}(s)) = \max_{\bar{w} \in U_p(s)} h_p(s, \bar{z}(s), p^0, \bar{w}), & s \in [0, 1] \text{ p.p.}, & (2.28) \\ g_p(\bar{z}(0), \bar{z}(1)) = 0, & & (2.29) \\ (\pi_p(\bar{z}), p^0) \neq 0, \quad p^0 \leq 0. & & (2.30) \end{cases}$$

### 2.3.4 Lien entre les conditions nécessaires d'optimalité

L'objectif est maintenant de montrer l'équivalence entre (NCOCP) et (NCPOCP), c'est-à-dire la proposition 2.3.7. Comme nous l'avons fait précédemment pour montrer l'équivalence entre les problèmes (OCP) et (POCP), nous allons utiliser les deux lemmes suivants, dont les preuves se trouvent dans l'annexe B. Afin de simplifier les notations, on introduit la notion d'égalité à une permutation près.

#### Définition 2.3.4 – Égalité à une permutation près

Soit  $u$  et  $v$  deux vecteurs de  $\mathbb{R}^k$ . On dit que  $u$  est égale à une permutation près à  $v$ , que l'on note  $u \cong v$ , s'il existe une matrice de permutation  $P$  de dimension  $k \times k$  telle que  $u = Pv$ .

**Remarque.** L'égalité à une permutation près est une relation d'équivalence. En effet, en prenant  $u, v, w \in \mathbb{R}^k$ , on peut montrer facilement que  $\cong$  est

- réflexive : en prenant  $P = \mathbf{I}_k$ , on a  $u = Pu$ .
- symétrique : Soit  $u \cong v$ . Par définition il existe une matrice de permutation  $P$  tels que  $u = Pv$ . Comme la matrice  $P$  est une matrice de permutation, elle est inversible, et  $P^{-1} = P^\top$ . On a donc  $v = P^\top u$ , ce qui montre que  $v \cong u$ .
- transitive : Soit  $u \cong v$  et  $v \cong w$ . Par définition, il existe deux matrices de permutations  $P$  et  $\bar{P}$  telles que  $u = Pv$  et  $v = \bar{P}w$ . La multiplication de deux matrices de permutation forme une matrice de permutation, donc la matrice  $\tilde{P} = P\bar{P}$  est une matrice de permutation. On a  $u = Pv = P\bar{P}w = \tilde{P}w$ , ce qui montre que  $u \cong w$ .



**Lemme 2.3.5 – Contraintes état/co-état**

Pour tout vecteur  $a = (a_0, \dots, a_N)$  et  $b = (b_0, \dots, b_N)$  de  $(\mathbb{R}^{2n})^{N+1}$ , on a

$$g_p(a, b) \cong \begin{pmatrix} b_0 - a_1 \\ \vdots \\ b_{N-1} - a_N \\ g(a_0, b_N) \end{pmatrix}.$$

► Preuve dans l'annexe B. ■

Pour un couple état/co-état  $\bar{z}(\cdot)$  du problème parallélisé, et en évaluant  $g_p$  en  $(\bar{z}(0), \bar{z}(1))$ , on remarque que la fonction de contraintes initiales et finales en état et co-état du problème parallélisé  $g_p$  correspond aux  $p$  conditions initiales et finales en état et co-état du problème (OCP) données par  $g$  entre  $\bar{z}_0(0)$  et  $\bar{z}_N(1)$  et à  $2nN$  conditions de raccordement en état et co-état aux interfaces, c'est-à-dire entre  $\bar{z}_i(1)$  et  $\bar{z}_{i+1}(0)$  pour tout  $i \in \mathbb{N}_{N-1}$ .

**Lemme 2.3.6 – Lien entre les pseudo-hamiltoniens**

Pour tout temps  $s \in \mathbb{R}$ , pour tout couple état/co-état parallèle  $\bar{z} \in (\mathbb{R}^{2n})^{N+1}$  et pour tout contrôle parallèle  $\bar{u} \in (\mathbb{R}^m)^{N+1}$ , on a

$$h_p(s, \bar{z}, p^0, \bar{u}) = \sum_{i=0}^N l_i h(\phi_i(s), \bar{z}_i, p^0, \bar{u}_i).$$

► Preuve dans l'annexe B. ■

Comme nous l'avons fait précédemment pour montrer l'équivalence entre les problèmes (OCP) et (POCP), on construit la fonction  $\Theta_p$  définie par

$$\Theta_p(z, p^0, u) = (\mathbf{S}_{\mathbf{p}, 2n}(z), p^0, \mathbf{S}_{\mathbf{p}, m}(u)).$$

**Proposition 2.3.7 – Équivalence entre (NCOCP) et (NCPOCP)**

Le point  $(z, p^0, u)$  est solution de (NCOCP) si et seulement si  $\Theta_p(z, p^0, u)$  est solution de (NCPOCP). On note cette relation par (NCOCP)  $\iff$  (NCPOCP).

► Soit  $(z, u)$  une solution de (NCOCP), avec  $z = (x, p)$ . On pose  $s \in [0, 1]$ ,

$$\bar{x} = \mathbf{S}_{\mathbf{p}, n}(x), \quad \bar{p} = \mathbf{S}_{\mathbf{p}, n}(p), \quad \bar{u} = \mathbf{S}_{\mathbf{p}, m}(u),$$

et on note  $\bar{z} = (\bar{x}, \bar{u})$ . En utilisant la définition de  $\mathbf{S}_{\mathbf{p}, \cdot}$ , on a directement  $(\bar{x}, \bar{u}) \in \mathcal{D}_p$  et  $p \in \text{AC}([0, 1], (\mathbb{R}^n)^{N+1})$ .

Le couple état/co-état parallèle  $\bar{z}$  satisfait l'équation (2.27) : pour tout  $i \in \mathbb{N}_N$ , en

utilisant la notation  $\bar{z}_i = (\bar{x}_i, \bar{p}_i)$ , on a

$$\begin{aligned}\dot{\bar{z}}_i(s) &= \phi'_i(s)(\dot{z} \circ \phi_i)(s) \\ &= l_i \mathbf{J}_{2n} \nabla_z h(\phi_i(s), \bar{z}_i(s), p^0, \bar{u}_i(s)) \\ &= l_i \begin{pmatrix} \nabla_p h(\phi_i(s), \bar{z}_i(s), p^0, \bar{u}_i(s)) \\ -\nabla_x h(\phi_i(s), \bar{z}_i(s), p^0, \bar{u}_i(s)) \end{pmatrix},\end{aligned}$$

ce qui, en réordonnant les éléments et en utilisant le lemme 2.3.6 nous donne

$$\begin{aligned}\dot{\bar{z}}(s) &= \begin{pmatrix} l_0 \nabla_p h(\phi_0(s), \bar{z}_0(s), p^0, \bar{u}_0(s)) \\ \vdots \\ l_N \nabla_p h(\phi_N(s), \bar{z}_N(s), p^0, \bar{u}_N(s)) \\ -l_0 \nabla_x h(\phi_0(s), \bar{z}_0(s), p^0, \bar{u}_0(s)) \\ \vdots \\ -l_N \nabla_x h(\phi_N(s), \bar{z}_N(s), p^0, \bar{u}_N(s)) \end{pmatrix} \\ &= \mathbf{J}_{2n(N+1)} \nabla_{\bar{z}} h_p(s, \bar{z}(s), \bar{u}(s)) \\ &= \vec{h}_p(s, \bar{z}(s), \bar{u}(s)).\end{aligned}$$

Le contrôle parallèle  $\bar{u}$  satisfait la condition de maximisation (2.28) du problème parallèle : en utilisant le lemme 2.3.6, on a directement

$$\begin{aligned}h_p(s, \bar{z}(s), p^0, \bar{u}(s)) &= \sum_{i=0}^N l_i h(\phi_i(s), \bar{z}_i(s), p^0, \bar{u}_i(s)) \\ &= \sum_{i=0}^N l_i h(\phi_i(s), (z \circ \phi_i)(s), p^0, (u \circ \phi_i)(s)) \\ &= \sum_{i=0}^N l_i \max_{w \in (U \circ \phi_i)(s)} h(\phi_i(s), (z \circ \phi_i)(s), p^0, w) \\ &= \max_{\bar{w} \in U_p(s)} h_p(s, \bar{z}(s), p^0, \bar{w}).\end{aligned}$$

Enfin, comme  $z \in \text{AC}([t_0, t_f], \mathbb{R}^{2n})$  et  $\bar{z} = \mathbf{S}_{\mathbf{p}, 2n}(z)$ , on a  $\bar{z}_i(1) = \bar{z}_{i+1}(0)$  pour tout  $i \in \mathbb{N}_{N-1}$ . En utilisant le lemme 2.3.5, on a directement  $g_p(\bar{z}(0), \bar{z}(1)) = 0$ .

La réciproque est montrée de la même manière, en remarquant que pour tout point admissible  $(\bar{z}, \bar{u})$  de (NCPOCP), il existe  $z$  et  $u$  tel que  $(\bar{z}, \bar{u}) = \Theta_p(z, u)$ . Plus précisément, on peut établir que  $z \in \text{AC}([t_0, t_f], \mathbb{R}^{2n})$  en utilisant

- l'appartenance de  $\bar{z}$  à  $\text{AC}([t_0, t_f], (\mathbb{R}^{2n})^{N+1})$ ,
- les conditions de raccordement en état/co-état (2.29) de (NCPOCP) et le lemme 2.3.5.

■

### 2.3.5 Schéma commutatif

Après avoir créé le problème parallélisé (**POCP**) à partir du problème (**OCP**), nous avons montré via la proposition 2.3.3 l'équivalence de ces deux problèmes. Nous avons ensuite montré dans la section 2.3.3 via la proposition 2.3.7 l'équivalence entre leurs conditions nécessaires d'optimalité, c'est-à-dire l'équivalence entre les deux problèmes (**NCOCP**) et (**NCPOCP**). Ces résultats sont résumés dans le graphe de la figure 2.2.

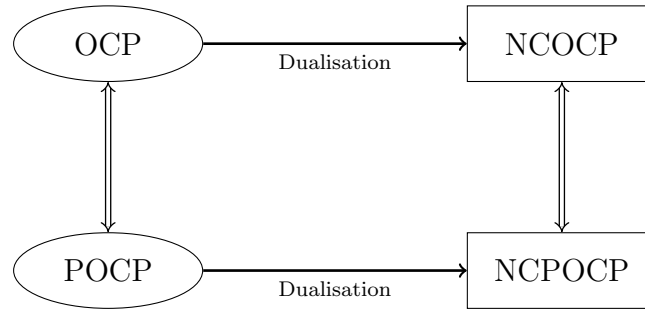


FIGURE 2.2 – Graphe commutatif montrant le lien entre les problèmes (**OCP**) et (**POCP**), ainsi que leurs conditions nécessaires d'optimalité respectives (**NCOCP**) et (**NCPOCP**). Les doubles flèches correspondent aux relations d'équivalence des propositions 2.3.3 et 2.3.7, et les flèches simples correspondent à l'opérateur de dualisation qui sera défini en fin de section 2.3.6 et qui correspond à l'application du principe du maximum.

### 2.3.6 Méthode numérique

Il est possible d'aller encore plus loin en montrant que la méthode de tir multiple appliquée au problème (**OCP**) est équivalente à la méthode de tir simple appliquée au problème (**POCP**), dans le sens de la proposition 2.3.8.

Sous l'hypothèse (**H2**), l'application exponentielle  $\exp_{\vec{h}_p}(s_2, s_1, \bar{z}_1, p^0)$  du champ de vecteur pseudo-hamiltonien  $h_p$  du problème (**POCP**) qui fournit la solution au temps  $s_2$  du problème

$$\begin{cases} \dot{\bar{z}}(t) = \vec{h}_p(s, \bar{z}(s), p^0, \bar{u}(s)), & s \in [s_1, s_2] \text{ p.p.}, \\ h_p(s, \bar{z}(s), p^0, u(s)) = \max_{\bar{w} \in U_p(s)} h_p(s, \bar{z}(s), p^0, \bar{w}), & s \in [s_1, s_2] \text{ p.p.} \\ \bar{z}(s_1) = \bar{z}_1, \end{cases}$$

est bien définie pour tous temps intermédiaires normalisés  $0 \leq s_1 < s_2 \leq 1$ , et pour tout état parallèle initial  $\bar{z}_1 \in (\mathbb{R}^{2n})^{N+1}$  tel que  $(\pi_p(\bar{z}_1), p^0) \neq 0$ . L'application de la méthode de tir simple sur le problème (**POCP**) mène donc à la résolution du problème aux deux bouts suivant

$$(TPBVPP) \quad \begin{cases} \bar{z}_f = \exp_{\vec{h}_p}(1, 0, \bar{z}_0, p^0), \\ g_p(\bar{z}_0, \bar{z}_f) = 0, \\ p^0 \leq 0, \quad (\pi_p(\bar{z}_0), p^0) \neq 0, \end{cases}$$

ce qui peut se faire en cherchant un zéro non trivial de la fonction de tir simple associée au problème parallélisé (POCP),  $S_p: (\mathbb{R}^{2n})^{N+1} \times \mathbb{R}^- \rightarrow (\mathbb{R}^{2n})^{N+1}$ , définie par

$$S_p(\bar{z}_0, p^0) = g_p(\bar{z}_0, \exp_{\bar{h}_p}^{\vec{r}}(1, 0, \bar{z}_0, p^0)).$$

**Proposition 2.3.8 – Tir simple sur (POCP) et tir multiple sur (OCP)**

Pour tout  $\bar{z} = (\bar{z}_0, \dots, \bar{z}_N) \in (\mathbb{R}^{2n})^{N+1}$  et pour tout  $p^0 \leq 0$ , on a

$$S_p(\bar{z}, p^0) \cong S_m(\bar{z}_0, \dots, \bar{z}_N, p^0).$$

► Soit  $\bar{z} = (\bar{z}_0, \dots, \bar{z}_N) \in (\mathbb{R}^{2n})^{N+1}$  et  $p^0 \leq 0$ . En utilisant des arguments similaires à ceux développés dans la preuve de la proposition 2.3.7, on peut établir que

$$\begin{aligned} \exp_{\bar{h}_p}^{\vec{r}}(1, 0, \bar{z}, p^0) &\cong \left( \exp_{\bar{h}}^{\vec{r}}(\phi_i(1), \phi_i(0), \bar{z}_i, p^0) \right)_{i \in \mathbb{N}_N} \\ &= \left( \exp_{\bar{h}}^{\vec{r}}(t_i, t_{i+1}, \bar{z}_i, p^0) \right)_{i \in \mathbb{N}_N} \end{aligned}$$

En utilisant ce résultat et le lemme 2.3.5, on obtient directement

$$\begin{aligned} S_p(\bar{z}, p^0) &= g_p(\bar{z}, \exp_{\bar{h}_p}^{\vec{r}}(1, 0, \bar{z}, p^0)) \\ &\cong \left( \begin{array}{c} \exp_{\bar{h}}^{\vec{r}}(t_1, t_0, \bar{z}_0, p^0) - \bar{z}_1 \\ \vdots \\ \exp_{\bar{h}}^{\vec{r}}(t_N, t_{N-1}, \bar{z}_{N-1}, p^0) - \bar{z}_N \\ g(\bar{z}_0, \exp_{\bar{h}}^{\vec{r}}(t_{N+1}, t_N, \bar{z}_N, p^0)) \end{array} \right) \\ &\cong S_m(\bar{z}_0, \dots, \bar{z}_N, p^0). \end{aligned}$$

■

Cette proposition nous montre que la méthode de tir multiple sur le problème (OCP) peut également être vue comme l'application de la méthode de tir simple sur le problème équivalent sous forme parallélisée (POCP). Afin de clarifier ce résultat, nous proposons d'introduire les opérateurs suivants :

- L'opérateur de *dualisation* fournit les conditions nécessaires d'optimalité associées à un problème donné. Cet opérateur, appliqué aux problèmes de commande optimale (OCP) et (POCP), conduit respectivement aux problèmes (NCOCP) et (NCPOCP) et, du point de vue des méthodes de tirs, aux problèmes (TPBVP) et (TPBVPP).
- L'opérateur de *découpage* transforme un problème aux deux bouts en un problème aux deux bouts avec conditions intermédiaires. Cet opérateur appliqué à (TPBVP) fournit (MPBVP), comme décrit dans la méthode de tir multiple à la section 2.2.2.

- L'opérateur de *parallélisation* transforme un problème de commande optimale en un problème de commande optimale parallélisé équivalent. Cet opérateur appliqué au problème (OCP) fournit le problème (POCP), comme décrit dans la section 2.3.1.

Les liens entre les méthodes numériques discutées ici sont schématisés par le graphe commutatif de la figure 2.3.

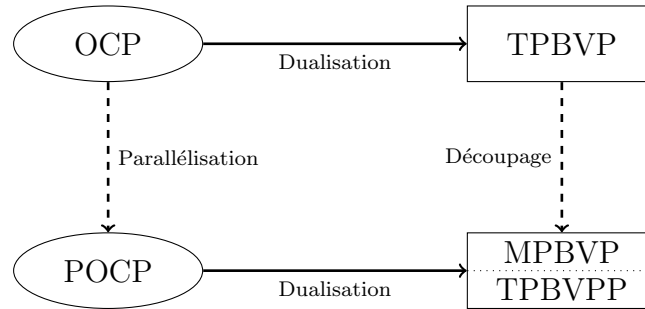


FIGURE 2.3 – Graphe commutatif des problèmes (OCP) et (MPBVP)/(TPBVPP).

## Conclusion

Ce chapitre introduit principalement le cadre de contrôle optimal dans lequel l'ensemble de ce travail s'inscrit. Plus précisément, nous avons introduit le problème de commande optimale (OCP) considéré. Ce problème sous forme de Lagrange est assez classique, et nous permet d'appliquer le principe du maximum qui nous fournit des conditions nécessaires d'optimalité.

Nous avons choisi de nous placer dans le cadre de l'hypothèse (H2). Bien qu'elle semble assez restrictive au premier abord, nous avons proposé deux cadres classiques dans laquelle elle est vérifiée, à savoir le cadre de Legendre strict et le cadre hamiltonien. Nous avons également prouvé au travers d'un exemple que cette hypothèse peut être vérifiée au-delà de ces deux cadres. Cette hypothèse nous permet d'écrire simplement les fonctions de tir intervenant dans les méthodes numériques indirectes, à savoir la méthode de tir simple et de tir multiple.

La dernière partie de ce chapitre propose un nouveau point de vue sur la méthode de tir multiple. En effet, cette méthode peut être vue comme l'application de la méthode de tir simple sur un problème dit parallélisé, à savoir le problème (POCP). Les principaux résultats de cette partie sont illustrés par les figures 2.2 et 2.3.

Cette contribution n'introduit pas de nouvelle méthode de résolution du problème de commande optimale (OCP). Le prochain chapitre introduit un nouveau chemin sur ces diagrammes, qui permettra le développement d'une nouvelle méthode de contrôle optimal.

## Méthode à deux niveaux

---

<b>3.1</b>	<b>Décomposition à deux niveaux</b> .....	<b>44</b>
3.1.1	Formulation à deux niveaux .....	44
3.1.2	Lien avec le problème initial .....	47
3.1.3	Conditions nécessaires d'optimalité .....	51
3.1.4	Lien entre les conditions nécessaires d'optimalité .....	52
3.1.5	Graphe commutatif .....	56
<b>3.2</b>	<b>Méthode numérique à deux niveaux</b> .....	<b>56</b>
3.2.1	Idée principale .....	57
3.2.2	Les problèmes (Macro) et (Micro) .....	57
3.2.3	Structure du problème (Macro) .....	59
3.2.4	Analyse de l'erreur .....	60
3.2.5	Avantages et inconvénients .....	62

Ce chapitre propose une nouvelle méthode de résolution de problèmes de commande optimale [21]. Elle est basée sur une formulation à deux niveaux du problème (OCP), et permet d'ajouter un nouveau chemin aux figures 2.2 et 2.3. Cette formulation à deux niveaux (BOCP) fait intervenir des fonctions valeur à des temps prédéterminés. La nouvelle méthode proposée, que nous appelons méthode *Macro-Micro*, est basée sur une approximation de ces fonctions valeur qui permet de transformer (BOCP) en deux problèmes imbriqués : un problème d'optimisation haut niveau dans lequel on cherche des états intermédiaires et des problèmes de contrôle optimal bas niveau dans lesquels on cherche les trajectoires d'état et de commande. Une telle formulation permet d'accélérer le processus de résolution et de diminuer le nombre de calculs tout en garantissant une solution proche de la solution optimale.

La formulation à deux niveaux (BOCP), sa relation avec (OCP) et les liens entre les conditions nécessaires d'optimalité de ces deux formulations, respectivement (NCOCP) et (NCBOCP), sont montrés dans la section 3.1. La nouvelle méthode de résolution de problème de commande optimale Macro-Micro est proposée dans la section 3.2.

### 3.1 Décomposition à deux niveaux

Dans la section 3.1.1, le problème (OCP) est écrit sous la forme d'un problème de commande optimale à deux niveaux : le problème (BOCP). Le lien entre ces deux formulations est donné dans la section 3.1.2. Nous formulons en section 3.1.3 les conditions nécessaires d'optimalité du problème à deux niveaux sous la forme du problème (NCBOCP). Nous établissons finalement en section 3.1.4 le lien entre ces conditions et celles du problème (OCP), c'est-à-dire le problème (NCOCP). Une synthèse de ces résultats est présentée en section 3.1.5.

#### 3.1.1 Formulation à deux niveaux

**Reformulation du problème de commande optimale.** On reprend ici le découpage de l'intervalle  $[t_0, t_f]$  en  $N + 1$  sous-intervalles  $(\Delta_i)_{i \in \mathbb{N}_N} = ([t_i, t_{i+1}])_{i \in \mathbb{N}_N}$ , introduit à la section 2.2.2, sur chacun desquels on définit le problème de commande optimale intermédiaire sous forme de Lagrange, avec une condition d'état initiale  $a \in \mathbb{R}^n$  et finale  $b \in \mathbb{R}^n$  fixées

$$(OCP_{i,a,b}) \quad \left\{ \begin{array}{l} V_i(a, b) = \min_{x, u} \int_{t_i}^{t_{i+1}} f^0(t, x(t), u(t)) dt \\ \text{s.c. } \dot{x}(t) = f(t, x(t), u(t)), \quad t \in \Delta_i \text{ p.p.}, \\ u(t) \in U(t), \quad t \in \Delta_i, \\ x(t_i) = a, \quad x(t_{i+1}) = b. \end{array} \right.$$

Soit  $i \in \mathbb{N}_N$ . La valeur  $V_i(a, b)$  correspond au coût de passage optimal de l'état  $a \in \mathbb{R}^n$  au temps initial  $t_i$  à l'état  $b \in \mathbb{R}^n$  au temps final  $t_{i+1}$ . La fonction  $V_i(\cdot, \cdot)$  est appelée la *fonction valeur* et correspond à une formulation de la fonction valeur de Bellman, centrale dans les équations de Hamilton-Jacobi-Bellman. La fonction valeur  $V_i(\cdot, \cdot)$  est définie sur l'ensemble des couples  $(a, b)$  tels que l'état final  $b \in \mathbb{R}^n$  soit atteignable au temps  $t_{i+1}$  à partir de l'état initial  $a \in \mathbb{R}^n$  au temps  $t_i$ . Pour pouvoir définir cet ensemble plus clairement, on propose de définir pour tout  $x_1 \in \mathbb{R}^n$  et  $t_1, t_2$  tels que  $t_0 \leq t_1 < t_2 \leq t_f$  :

- la fonction  $x(t_2, t_1, x_1, u)$  comme la solution au temps  $t_2$  du problème de Cauchy suivant

$$\dot{x}(t) = f(t, x(t), u(t)), \quad t \in [t_1, t_2] \text{ p.p.}, \quad x(t_1) = x_1;$$

- l'ensemble des *contrôles admissibles*  $\mathcal{U}_{t_2, t_1, x_1}$  sur l'intervalle  $[t_1, t_2]$  par l'ensemble des contrôles  $u \in L^\infty([t_1, t_2], \mathbb{R}^m)$  tels que pour tout  $t \in [t_1, t_2]$ ,  $u(t) \in U(t)$  et tels que  $x(t_2, t_1, x_1, u)$  soit bien défini ;
- l'*application entrée/sortie*  $E_{t_2, t_1, x_1} : \mathcal{U}_{t_2, t_1, x_1} \rightarrow \mathbb{R}^n$  par

$$E_{t_2, t_1, x_1}(u) = x(t_2, t_1, x_1, u);$$

- l'*ensemble accessible*  $\mathcal{A}(t_2, t_1, x_1)$  au temps  $t_2$  à partir de l'état  $x_1$  au temps  $t_1$  par

$$\mathcal{A}(t_2, t_1, x_1) = E_{t_2, t_1, x_1}(\mathcal{U}_{t_2, t_1, x_1}).$$

Finalement, la fonction valeur  $V_i$  est définie sur

$$\Omega_i = \left\{ (a, b) \mid b \in \mathcal{A}(t_{i+1}, t_i, a) \right\}.$$

**Remarque.** On admet implicitement que pour tout  $i \in \mathbb{N}_N$  et pour tout  $(a, b) \in \Omega_i$ , il existe  $(x_i, u_i)$  solution de  $(\text{OCP}_{i,a,b})$ . Autrement dit, on considère que le minimum n'est pas un infimum.

En dénotant  $\mathcal{D}_i = \text{AC}(\Delta_i, \mathbb{R}^n) \times L^\infty(\Delta_i, \mathbb{R}^m)$ , le coût  $J_i(a, b): \mathcal{D}_i \rightarrow \mathbb{R}$  du problème de commande optimale intermédiaire  $(\text{OCP}_{i,a,b})$  est défini par

$$J_i(x, u) = \int_{t_i}^{t_{i+1}} f^0(t, x(t), u(t)) dt.$$

On note  $\mathcal{F}_i(a, b)$  l'ensemble des points admissibles de  $(\text{OCP}_{i,a,b})$ , c'est-à-dire

$$\mathcal{F}_i(a, b) = \left\{ (x, u) \in \mathcal{D}_i \mid \begin{array}{l} \dot{x}(t) = f(t, x(t), u(t)), \quad t \in [t_i, t_{i+1}] \text{ p.p.}, \\ u(t) \in U(t) \quad t \in [t_i, t_{i+1}], \\ x(t_i) = a, \quad x(t_{i+1}) = b \end{array} \right\}$$

et  $\mathcal{S}_i(a, b)$  l'ensemble des solutions de  $(\text{OCP}_{i,a,b})$ , c'est-à-dire

$$\mathcal{S}_i(a, b) = \arg \min_{(x,u) \in \mathcal{F}_i(a,b)} J_i(a, b).$$

La formulation à deux niveaux  $(\text{BOCP})$  du problème de commande optimale  $(\text{OCP})$  est donnée par

$$(\text{BOCP}) \quad \begin{cases} \min_X V(X) = \sum_{i=0}^N V_i(X_i, X_{i+1}), \\ \text{s.c. } c(X_0, X_{N+1}) = 0, \quad X \in \mathcal{X}, \end{cases}$$

où l'ensemble  $\mathcal{X} \subset (\mathbb{R}^n)^{N+2}$  correspond à l'ensemble des états intermédiaires accessibles et est défini par

$$\mathcal{X} = \left\{ (X_0, \dots, X_{N+1}) \in (\mathbb{R}^n)^{N+2} \mid \forall i \in \mathbb{N}_N, \quad (X_i, X_{i+1}) \in \Omega_i \right\}.$$

L'objectif est donc de trouver le vecteur d'états intermédiaires  $X \in \mathcal{X}$  admissible, c'est-à-dire tel que  $X \in \mathcal{X}$  et  $c(X_0, X_{N+1}) = 0$ , qui minimise  $V$  parmi tous les autres vecteurs d'états intermédiaires admissibles.

**Lien avec l'optimisation à deux niveaux.** Nous allons montrer dans ce paragraphe que le problème  $(\text{BOCP})$ , qui se formule comme un problème d'optimisation, est également un problème de commande optimale à deux niveaux.

D'après [15], un problème à deux niveaux, ou *Leader-Follower*, peut se formuler de la manière suivante :

$$(L) \quad \begin{cases} \min_x F(x, y), \\ \text{s.c. } x \in X, \quad y \in S(x), \end{cases}$$



où  $S(x)$  est l'ensemble des solutions de

$$(F) \quad \begin{cases} \min_y f(x, y), \\ y \in Y(x). \end{cases}$$

On remarque que le problème (L) peut être mal défini lorsque  $S(x)$  contient plusieurs solutions. Dans ce cas, quelle que soit la valeur de  $y$  choisie dans  $S(x)$ , la valeur optimale du problème (L) sera au mieux celle du problème optimiste

$$(L_o) \quad \begin{cases} \min_x \min_y F(x, y), \\ \text{s.c. } x \in X, \quad y \in S(x), \end{cases}$$

et au pire celle du problème pessimiste

$$(L_p) \quad \begin{cases} \min_x \max_y F(x, y), \\ \text{s.c. } x \in X, \quad y \in S(x). \end{cases}$$

Le cas des *Single-Leader-Multi-Followers Games* (SLMFG), issu de la théorie des jeux [26], nous intéresse plus spécifiquement. Il s'agit d'un cas particulier pour lequel l'ensemble  $S(x)$  correspond à l'ensemble des  $y = (y_0, \dots, y_N) \in \mathbb{R}^m$  tels que pour tout entier  $i \in \mathbb{N}_N$ , le point  $y_i \in \mathbb{R}^k$  (avec  $k(N+1) = m$ ) est solution de

$$(F_{i,x,y_{-i}}) \quad \begin{cases} \min_{y_i} f_i(x, y_i, y_{-i}), \\ \text{s.c. } y_i \in Y_i(x, y_{-i}), \end{cases}$$

où  $y_{-i}$  correspond au vecteur composé de tous les éléments de  $y$  sauf  $y_i$ , c'est-à-dire qu'on a  $y_{-i} = (y_0, \dots, y_{i-1}, y_{i+1}, \dots, y_N) \in \mathbb{R}^{m-k}$ .

Imaginons maintenant un problème d'optimisation à deux niveaux sous la forme d'un SLMFG encore plus particulier où les problèmes  $(F_{i,x,y_{-i}})$  sont indépendants, et où  $f_i$  ne dépend pas de  $x$

$$(F_{i,x}) \quad \begin{cases} V_i(x) = \min_{y_i} f_i(y_i), \\ \text{s.c. } y_i \in Y_i(x). \end{cases}$$

On suppose également que  $F$  ne dépend pas de la variable  $x$  et s'écrit

$$F(y) = \sum_{i=0}^N f_i(y_i).$$

Le problème (L) se formule alors de la manière suivante

$$\begin{cases} \min_x \min_y \sum_{i=0}^N f_i(y_i), \\ \text{s.c. } x \in X, \\ \forall i \in \mathbb{N}_N, \quad y_i \in S_i(x), \end{cases} \quad \text{ou} \quad \begin{cases} \min_x \max_y \sum_{i=0}^N f_i(y_i), \\ \text{s.c. } x \in X, \\ \forall i \in \mathbb{N}_N, \quad y_i \in S_i(x), \end{cases}$$

sous sa forme optimiste (à gauche) ou pessimiste (à droite), correspondant respectivement aux problèmes  $(L_o)$  et  $(L_p)$ . Pour tout  $i \in \mathbb{N}_N$ , l'ensemble  $S_i(x)$  correspond à l'ensemble des solutions du problème  $(F_{i,x})$ . Par définition, pour tout  $x \in X$ , le coût de toutes les solutions  $y_i \in S_i(x)$  est le même, à savoir  $f_i(y_i) = V_i(x)$ , et en particulier, on a pour tout  $i \in \mathbb{N}_N$  et pour tout  $x \in X$

$$\min_{y_i \in S_i(x)} f_i(y_i) = \max_{y_i \in S_i(x)} f_i(y_i) = V_i(x).$$

En définissant l'ensemble  $S(x)$  par  $S(x) = \{y \mid \forall i \in \mathbb{N}_N, y_i \in S_i(x)\}$ , on a directement que

$$\min_{y \in S(x)} \sum_{i=0}^N f_i(y_i) = \max_{y \in S(x)} \sum_{i=0}^N f_i(y_i) = \sum_{i=0}^N V_i(x).$$

Les problèmes à deux niveaux sous forme optimiste  $(L_o)$  et pessimiste  $(L_p)$  sont donc équivalents et peuvent être formulés sous la forme suivante

$$(BOP) \quad \begin{cases} \min_x \sum_{i=0}^N V_i(x) \\ \text{s.c. } x \in X. \end{cases}$$

On remarque que la variable  $y$  a disparu dans cette formulation. Il est toutefois possible de retrouver ce point en prenant n'importe quel élément  $y$  de  $S(x)$ .

Finalement, le problème  $(BOCP)$  est de la même forme que le problème  $(BOP)$ , il est donc bien un problème à deux niveaux. Par ailleurs, comme les  $V_i(X_i, X_{i+1})$  qui interviennent dans  $(BOCP)$  sont les valeurs optimales de problèmes de commande optimale  $(OCP_{i,a,b})$  où  $a = X_i$  et  $b = X_{i+1}$ , le problème  $(BOCP)$  est un problème de commande optimale à deux niveaux, ce qui justifie son appellation.

### 3.1.2 Lien avec le problème initial

Après avoir dérivé le problème  $(BOCP)$  du problème  $(OCP)$ , nous allons montrer qu'ils sont équivalents au sens des propositions 3.1.3 et 3.1.4. Avant cela, nous proposons d'établir deux résultats préalables.

#### Lemme 3.1.1

Si  $(x, u)$  est un point admissible de  $(OCP)$  alors pour tout  $i \in \mathbb{N}_N$ , le couple  $(\bar{x}_i, \bar{u}_i) = (x|_{\Delta_i}, u|_{\Delta_i})$  est un point admissible de  $(OCP_{i,x(t_i),x(t_{i+1})})$ . De plus, on a

$$J(x, u) = \sum_{i=0}^N J_i(\bar{x}_i, \bar{u}_i).$$

**Remarque.** Pour tout  $u \in L^\infty(\mathbb{R}, \mathbb{R}^k)$ , et pour tout  $E \subset \mathbb{R}$ , la notation  $u|_E$  correspond à la restriction de la fonction  $u$  sur  $E$ .

► Soit  $(x, u)$  un point admissible du problème (OCP) et  $i \in \mathbb{N}_N$ . On pose

$$\bar{x}_i = x|_{\Delta_i} \quad \text{et} \quad \bar{u}_i = u|_{\Delta_i}.$$

Comme  $(x, u)$  est un point admissible du problème (OCP), on a

$$\left. \begin{array}{l} \dot{x}(t) = f(t, x(t), u(t)), \\ u(t) \in U(t), \quad t \in [t_0, t_f] \end{array} \right\} \implies \left\{ \begin{array}{l} \dot{\bar{x}}_i(t) = f(t, \bar{x}_i(t), \bar{u}_i(t)), \\ \bar{u}_i(t) \in U(t), \quad t \in \Delta_i \end{array} \right.$$

De plus, nous avons trivialement  $x(t_i) = \bar{x}_i(t_i)$  et  $x(t_{i+1}) = \bar{x}_i(t_{i+1})$ . Le point  $(\bar{x}_i, \bar{u}_i)$  est donc un point admissible du problème (OCP $_{i, x(t_i), x(t_{i+1})}$ ). De plus, nous avons directement

$$J(x, u) = \int_{t_0}^{t_f} f^0(t, x(t), u(t)) \, dt = \sum_{i=0}^N \int_{t_i}^{t_{i+1}} f^0(t, \bar{x}_i(t), \bar{u}_i(t)) \, dt = \sum_{i=0}^N J_i(\bar{x}_i, \bar{u}_i).$$

■

### Corollaire 3.1.2

Si  $(x, u)$  est une solution de (OCP), alors pour tout  $i \in \mathbb{N}_N$ , le point  $(x|_{\Delta_i}, u|_{\Delta_i})$  est une solution de (OCP $_{i, x(t_i), x(t_{i+1})}$ ).

► Soit  $(x^*, u^*)$  une solution de (OCP). On pose pour tout  $i \in \mathbb{N}_N$

$$\bar{x}_i^* = x^*|_{\Delta_i} \quad \text{et} \quad \bar{u}_i^* = u^*|_{\Delta_i}.$$

D'après le lemme 3.1.1, pour tout  $i \in \mathbb{N}_N$ , le couple  $(\bar{x}_i^*, \bar{u}_i^*)$  est un point admissible du problème (OCP $_{i, x^*(t_i), x^*(t_{i+1})}$ ). Raisonnons par l'absurde et supposons qu'il existe  $j \in \mathbb{N}_N$  tel que le point  $(\bar{x}_j^*, \bar{u}_j^*)$  ne soit pas une solution de (OCP $_{j, x^*(t_j), x^*(t_{j+1})}$ ), c'est-à-dire qu'il existe un point admissible  $(\bar{x}_j, \bar{u}_j)$  de (OCP $_{j, x^*(t_j), x^*(t_{j+1})}$ ) tel que  $J_j(\bar{x}_j, \bar{u}_j) < J_j(\bar{x}_j^*, \bar{u}_j^*)$ . On construit le point  $(x, u)$  tel que pour tout  $t \in [t_0, t_f]$

$$x(t) = \begin{cases} \bar{x}_j(t) & \text{si } t \in \Delta_j, \\ \bar{x}_i^*(t) & \text{sinon,} \end{cases} \quad \text{et} \quad u(t) = \begin{cases} \bar{u}_j(t) & \text{si } t \in \Delta_j, \\ \bar{u}_i^*(t) & \text{sinon.} \end{cases}$$

Tout d'abord, l'état  $x$  est bien une fonction absolument continue car d'une part les fonctions  $\bar{x}_j$  et  $\bar{x}_i^*$  sont des fonctions absolument continues sur leurs intervalles de temps respectifs, pour tout  $i \in \mathbb{N}_N$  et d'autre part, elles se raccordent aux extrémités des intervalles  $\Delta_i$  grâce aux contraintes d'état initiale et finale de (OCP $_{i, x^*(t_i), x^*(t_{i+1})}$ ). De plus, l'état  $x$  satisfait la dynamique d'état sur  $[t_0, t_f]$  car pour tout  $i \in \mathbb{N}_N$ , l'état  $\bar{x}_i^*$  satisfait la dynamique d'état sur  $\Delta_i$  (de même pour l'état  $\bar{x}_j$  sur  $\Delta_j$ ). De la même manière, le contrôle  $u$  satisfait la contrainte de contrôle. L'état  $x$  satisfait les contraintes aux bords

$$c(x(t_0), x(t_f)) = c(x^*(t_0), x^*(t_f)) = 0$$

car  $(x^*, u^*)$  est un point admissible de (OCP). Finalement, le couple  $(x, u)$  est un

point admissible du problème (OCP). D'après le lemme 3.1.1

$$J(x, u) = J_j(\bar{x}_j, \bar{u}_j) + \sum_{\substack{i=0 \\ i \neq j}}^N J_i(\bar{x}_i^*, \bar{u}_i^*) < \sum_{i=0}^N J_i(\bar{x}_i^*, \bar{u}_i^*) = J(x^*, u^*),$$

ce qui est impossible car le point  $(x^*, u^*)$  est une solution du problème (OCP). ■

On définit maintenant la fonction  $\theta_b: \mathcal{D} \rightarrow (\mathbb{R}^n)^{N+2}$  par

$$\theta_b(x, u) = (x(t_i))_{i \in \mathbb{N}_{N+1}}$$

qui à un couple état/contrôle  $(x, u)$  associe le vecteur composé des valeurs de l'état  $x$  aux temps  $t_i$ , pour tout  $i \in \mathbb{N}_{N+1}$ .

### Proposition 3.1.3 – Équivalence des solutions (Partie 1)

Si le point  $(x, u)$  est une solution de (OCP), alors le point  $\theta_b(x, u)$  est une solution de (BOCP).

► Soit  $(x^*, u^*)$  une solution du problème (OCP), c'est-à-dire que le point  $(x^*, u^*)$  est un point admissible de (OCP) et que  $(x^*, u^*)$  minimise  $J$  parmi tous les autres points admissibles de (OCP). On pose  $X^* = \theta_b(x^*, u^*)$ , c'est-à-dire que pour tout  $i \in \mathbb{N}_{N+1}$ , on a  $X_i^* = x^*(t_i)$ . Pour tout  $i \in \mathbb{N}_N$ , l'état  $X_{i+1}^* = x^*(t_{i+1})$  est atteignable au temps  $t_{i+1}$  à partir de l'état  $X_i^* = x^*(t_i)$  au temps  $t_i$ . En effet, dans le cas contraire, le point  $(x^*, u^*)$  ne serait pas admissible pour le problème (OCP). Donc

$$\forall i \in \mathbb{N}_N, \quad (X_i^*, X_{i+1}^*) \in \Omega_i, \quad \implies \quad X^* \in \mathcal{X}.$$

De plus, comme  $(x^*, u^*)$  est un point admissible du problème (OCP), on a

$$c(X_0^*, X_{N+1}^*) = c(x^*(t_0), x^*(t_f)) = 0.$$

Enfin, d'après le corollaire 3.1.2, pour tout  $i \in \mathbb{N}_N$ , le point  $(\bar{x}_i^*, \bar{u}_i^*) = (x^*|_{\Delta_i}, u^*|_{\Delta_i})$  est une solution du problème (OCP $_{i, X_i^*, X_{i+1}^*}$ ), c'est-à-dire que  $(\bar{x}_i^*, \bar{u}_i^*)$  est un point admissible et que  $J_i(\bar{x}_i^*, \bar{u}_i^*) = V_i(X_i^*, X_{i+1}^*)$ .

Montrons maintenant que  $X^*$  minimise  $V$  parmi tous les points admissibles du problème (BOCP). Pour cela, raisonnons par l'absurde et supposons qu'il existe un vecteur d'états intermédiaires  $X \in (\mathbb{R}^n)^{N+2}$  tel que

$$X \in \mathcal{X}, \quad c(X_0, X_{n+1}) = 0 \quad \text{et} \quad V(X) < V(X^*).$$

Par définition,  $V(X) = \sum_{i=0}^N V_i(X_i, X_{i+1})$ , où  $V_i(X_i, X_{i+1})$  correspond au coût optimal du problème (OCP $_{i, X_i, X_{i+1}}$ ). Il existe donc une solution  $(\bar{x}_i, \bar{u}_i)$  du problème (OCP $_{i, X_i, X_{i+1}}$ ) pour tout  $i \in \mathbb{N}_N$ , c'est-à-dire que

$$(\bar{x}_i, \bar{u}_i) \text{ admissible de (OCP}_{i, X_i, X_{i+1}}) \quad \text{et} \quad J_i(\bar{x}_i, \bar{u}_i) = V_i(X_i, X_{i+1}).$$

On construit  $(x, u)$  tel que pour tout  $i \in \mathbb{N}_N$  et pour tout  $t \in \Delta_i$ ,

$$x(t) = \bar{x}_i(t) \quad \text{et} \quad u(t) = \bar{u}_i(t).$$

Ce couple est admissible pour le problème (OCP) car d'une part pour tout  $i \in \mathbb{N}_N$ , le point  $(\bar{x}_i, \bar{u}_i)$  est admissible pour  $(\text{OCP}_{i, X_i, X_{i+1}})$  et d'autre part  $c(x(t_0), x(t_f)) = 0$  puisque  $X$  est admissible pour le problème (BOCP). De plus, on a

$$\begin{aligned} J(x, u) &= \sum_{i=0}^N J_i(\bar{x}_i, \bar{u}_i) = \sum_{i=0}^N V_i(X_i, X_{i+1}) = V(X) \\ &< V(X^*) = \sum_{i=0}^N V_i(X_i^*, X_{i+1}^*) = \sum_{i=0}^N J_i(\bar{x}_i^*, \bar{u}_i^*) = J(x^*, u^*), \end{aligned}$$

ce qui est impossible car  $(x^*, u^*)$  est une solution du problème (OCP).  $\blacksquare$

### Proposition 3.1.4 – Équivalence des solutions (Partie 2)

Si le point  $X$  est une solution de (BOCP), alors il existe une solution  $(x, u)$  de (OCP) tels que  $\theta_b(x, u) = X$ .

► Soit  $X^*$  une solution de (BOCP), c'est-à-dire que le vecteur  $X^*$  est admissible

$$X^* \in \mathcal{X}, \quad c(X_0^*, X_{N+1}^*) = 0,$$

et que  $X^*$  minimise  $V$  parmi tous les autres points admissibles de (BOCP). Comme le vecteur  $X^* \in \mathcal{X}$ , l'état  $X_{i+1}^*$  est atteignable au temps  $t_{i+1}$  à partir de l'état  $X_i^*$  au temps  $t_i$ , pour tout  $i \in \mathbb{N}_N$ . De plus, comme

$$V(X^*) = \sum_{i=0}^N V_i^*(X_i^*, X_{i+1}^*),$$

où  $V_i(X_i^*, X_{i+1}^*)$  est le coût optimal du problème  $(\text{OCP}_{i, X_i^*, X_{i+1}^*})$ , il existe une solution  $(\bar{x}_i^*, \bar{u}_i^*)$  du problème  $(\text{OCP}_{i, X_i^*, X_{i+1}^*})$  pour tout  $i \in \mathbb{N}_N$ , c'est-à-dire que

$$(\bar{x}_i^*, \bar{u}_i^*) \text{ admissible de } (\text{OCP}_{i, X_i^*, X_{i+1}^*}) \quad \text{et} \quad V(X_i^*, X_{i+1}^*) = J_i(\bar{x}_i^*, \bar{u}_i^*).$$

On construit le point  $(x^*, u^*)$  tel que pour tout  $i \in \mathbb{N}_N$ , et pour presque tout  $t \in \Delta_i$ ,

$$x^*(t) = \bar{x}_i^*(t), \quad \text{et} \quad u^*(t) = \bar{u}_i^*(t).$$

On remarque tout d'abord qu'on a par construction  $X^* = \theta_b(x^*, u^*)$ . De plus, le couple  $(x^*, u^*)$  est admissible pour le problème (OCP) car d'une part pour tout  $i \in \mathbb{N}_N$ , le point  $(\bar{x}_i^*, \bar{u}_i^*)$  est un point admissible de  $(\text{OCP}_{i, X_i^*, X_{i+1}^*})$  et d'autre part  $c(x^*(t_0), x^*(t_f)) = c(X_0^*, X_{N+1}^*) = 0$  puisque  $X^*$  est admissible pour le problème (BOCP).

Montrons maintenant que  $(x^*, u^*)$  minimise le coût  $J$  parmi tous les points admissibles du problème (OCP). Pour cela, raisonnons par l'absurde et supposons qu'il existe un point admissible  $(x, u)$  du problème (OCP) tel que  $J(x, u) < J(x^*, u^*)$ . On pose  $X = \theta_b(x, u)$  et pour tout  $i \in \mathbb{N}_N$ , on pose  $(\bar{x}_i, \bar{u}_i) = (x|_{\Delta_i}, u|_{\Delta_i})$ .

D'après le lemme 3.1.1, le point  $(\bar{x}_i, \bar{u}_i)$  est un point admissible de  $(\text{OCP}_{i, X_i, X_{i+1}})$ , et  $J(x, u) = \sum_{i=0}^N J_i(\bar{x}_i, \bar{u}_i)$ . On a donc

$$\begin{aligned} V(X) &= \sum_{i=0}^n V_i(X_i, X_{i+1}) \leq \sum_{i=0}^N J_i(\bar{x}_i, \bar{u}_i) = J(x, u) \\ &< J(x^*, u^*) = \sum_{i=0}^N J_i(\bar{x}_i^*, \bar{u}_i^*) = \sum_{i=0}^n V_i(X_i^*, X_{i+1}^*) = V(X^*), \end{aligned}$$

ce qui est impossible car  $X^*$  est une solution de  $(\text{BOCP})$ . ■

À la différence 3.1.3, la proposition 3.1.4 n'est pas constructive. Comme nous l'avons vu dans la preuve précédente, il est cependant assez simple de retrouver une solution  $(x, u)$  à partir du vecteur d'états intermédiaires  $X = (X_0, \dots, X_{N+1})$  solution de  $(\text{BOCP})$ . En effet, on peut pour cela choisir n'importe quelle solution  $(x_i, u_i) \in \mathcal{S}_i(X_i, X_{i+1})$  de  $(\text{OCP}_{i, X_i, X_{i+1}})$  pour tout  $i \in \mathbb{N}_N$  et construire le couple  $(x, u)$  solution du problème  $(\text{OCP})$  en posant pour tout  $t \in [t_0, t_f]$ ,

$$x(t) = x_i(t) \quad \text{et} \quad u(t) = u_i(t) \quad \text{quand} \quad t \in \Delta_i.$$

Finalement, à travers les propositions 3.1.3 et 3.1.4, nous venons de montrer l'équivalence entre les solutions des problèmes  $(\text{OCP})$  et  $(\text{BOCP})$ . Nous proposons de noter cette relation par la relation d'équivalence classique  $(\text{OCP}) \iff (\text{BOCP})$ .

### 3.1.3 Conditions nécessaires d'optimalité

Nous venons de montrer le lien entre les solutions de  $(\text{OCP})$  et  $(\text{BOCP})$ . Nous voulons maintenant écrire des conditions nécessaires d'optimalité de  $(\text{BOCP})$  et établir leur lien avec celles de  $(\text{OCP})$ , c'est-à-dire  $(\text{NCOCP})$ .

**Hypothèses.** Avant de voir quelles sont les conditions nécessaires d'optimalité du problème  $(\text{BOCP})$ , nous allons d'abord introduire les trois hypothèses suivantes.

**Hypothèse H5** (Différentiabilité des fonctions valeur).

Pour tout  $X$  solution de  $(\text{BOCP})$ , la fonction  $V$  est différentiable en  $X$ .

Les fonctions valeur sont solutions des équations aux dérivées partielles de Hamilton-Jacobi-Bellman, et elles peuvent être non différentiables pour certains problèmes. L'hypothèse **(H5)** est donc restrictive. Nous la conservons néanmoins, pour rester dans un cadre simple et nous focaliser sur la méthodologie, tout en gardant à l'esprit que les résultats suivants pourraient être généralisés à un cadre plus large, en utilisant des outils tels que la théorie des solutions de viscosité ou l'analyse non lisse décrits dans [13].

**Hypothèse H6** (Normalité des BC-extrémales).

Pour tout  $X = (X_0, \dots, X_{N+1})$  solution du problème  $(\text{BOCP})$ , les BC-extrémales associées aux solutions de  $(\text{OCP}_{i, X_i, X_{i+1}})$  sont normales pour tout  $i \in \mathbb{N}_N$ .

**Hypothèse H7.**

Si  $X$  est solution du problème (BOCP), alors  $X \in \overset{\circ}{\mathcal{X}}$ , où  $\overset{\circ}{\mathcal{X}}$  correspond à l'intérieur de l'ensemble  $\mathcal{X}$ .

On pourra se convaincre par la suite avec l'interprétation géométrique du co-état décrit dans la section 4.2.2 que (H6)  $\implies$  (H7). Cependant, dans un objectif de simplicité, on ne montrera pas ce résultat, et on considère donc ces deux hypothèses.

On remarque que sous l'hypothèse (H7), si  $X = (X_0, \dots, X_{N+1})$  est une solution du problème (BOCP), alors pour tout  $i \in \mathbb{N}_N$ , on a  $(X_i, X_{i+1}) \in \overset{\circ}{\Omega}_i$ , ou autrement dit,  $X_{i+1} \in \overset{\circ}{\mathcal{A}}(t_{i+1}, t_i, X_i)$ .

**Conditions nécessaires d'optimalité.** Sous l'hypothèse (H7), le problème (BOCP) peut se réécrire de la manière suivante

$$(BOCP) \quad \begin{cases} \min_X V(X) = \sum_{i=0}^N V_i(X_i, X_{i+1}), \\ \text{s.c. } c(X_0, X_{N+1}) = 0, \quad X \in \overset{\circ}{\mathcal{X}}. \end{cases}$$

Il s'agit d'un problème d'optimisation dans lequel l'ensemble  $\overset{\circ}{\mathcal{X}}$  est ouvert. Le lagrangien  $L : (\mathbb{R}^n)^{N+2} \times \mathbb{R}^p \rightarrow \mathbb{R}$  associé à ce problème s'écrit donc

$$L(X, \lambda) = V(X) - (\lambda \mid c(X_0, X_{N+1})).$$

En utilisant les conditions de Karush-Kuhn-Tucker, sous l'hypothèse (H5), si  $X$  est solution de (BOCP), il existe  $\lambda \in \mathbb{R}^p$  tel que le couple  $(X, \lambda)$  soit solution de

$$(NCBOCP) \quad \begin{cases} \nabla_X L(X, \lambda) = 0, \\ X \in \overset{\circ}{\mathcal{X}}, \quad c(X_0, X_{N+1}) = 0. \end{cases}$$

En utilisant la définition des fonctions  $L$  et  $V$ , ces conditions se réécrivent sous la forme suivante

$$\begin{cases} (\nabla_a V_0(X_0, X_1), \nabla_b V_N(X_N, X_{N+1})) - c'(X_0, X_{N+1})^\top \lambda = 0, \\ \nabla_b V_{i-1}(X_{i-1}, X_i) + \nabla_a V_i(X_i, X_{i+1}) = 0, \quad \forall i \in \{1, \dots, N\}, \\ c(X_0, X_{N+1}) = 0, \quad X \in \overset{\circ}{\mathcal{X}}. \end{cases}$$

Nous voyons apparaître dans les deux premières lignes de ce problème le gradient des fonctions valeur par rapport aux conditions initiale  $a$  et finale  $b$ .

**3.1.4 Lien entre les conditions nécessaires d'optimalité**

Afin d'aller plus loin, nous devons utiliser le lien établi dans le théorème suivant entre les gradients des fonctions valeur et les co-états du principe du maximum.

**Théorème 3.1.5 – Gradient des fonctions valeur et co-états**

Pour  $(a, b) \in \mathring{\Omega} := \{(a, b) \mid b \in \mathring{A}(t_f, t_0, a)\}$ , on considère le problème de commande optimale suivant

$$(\text{OCP}_*) \quad \left\{ \begin{array}{l} V_*(a, b) = \min_{x, u} \int_{t_0}^{t_f} f^0(t, x(t), u(t)) \, dt, \\ \text{s.c. } \dot{x}(t) = f(t, x(t), u(t)) \quad t \in [t_0, t_f] \text{ p.p.}, \\ u(t) \in U(t), \quad t \in [t_0, t_f], \\ x(t_0) = a, \quad x(t_f) = b. \end{array} \right.$$

On suppose que la fonction valeur  $V_*$  est différentiable au point  $(a, b)$  et que les BC-extrémales associées aux solutions du problème  $(\text{OCP}_*)$  sont normales. Sous ces hypothèses, si  $(x, u)$  est une solution de  $(\text{OCP}_*)$  avec  $(x, p, -1, u)$  une BC-extrémale associée, alors

$$\nabla V_*(x(t_0), x(t_f)) = (-p(t_0), p(t_f)).$$

**Remarque.** On remarque que  $(\text{OCP}_*)$  est un cas particulier du problème  $(\text{OCP})$ , où l'état initial et final sont fixés :  $c(x(t_0), x(t_f)) = (x(t_0) - a, x(t_f) - b)$ .

► On veut prouver les deux équations suivantes

$$\nabla_a V_*(x(t_0), x(t_f)) = -p(t_0), \quad (3.1)$$

$$\nabla_b V_*(x(t_0), x(t_f)) = p(t_f). \quad (3.2)$$

L'équation (3.1) est un résultat classique [6, 11, 14, 20, 60] sous les hypothèses considérées. Nous allons maintenant utiliser ce résultat pour prouver l'équation (3.2). Pour ce faire, on transforme  $(\text{OCP}_*)$  en  $(\text{ROCP}_*)$  en utilisant le changement de variable (inversion du temps)  $\phi: [t_0, t_f] \rightarrow [t_0, t_f]$  défini par  $\phi(t) = t_f + t_0 - t$  :

$$(\text{ROCP}_*) \quad \left\{ \begin{array}{l} V_R(b, a) = \min_{\hat{x}, \hat{u}} \int_{t_0}^{t_f} f^0(\phi(t), \hat{x}(t), \hat{u}(t)) \, dt, \\ \text{s.c. } \dot{\hat{x}}(t) = -f(\phi(t), \hat{x}(t), \hat{u}(t)), \quad t \in [t_0, t_f] \text{ p.p.}, \\ \hat{u}(t) \in U(\phi(t)), \quad t \in [t_0, t_f], \\ \hat{x}(t_0) = b, \quad \hat{x}(t_f) = a. \end{array} \right.$$

Nous avons naturellement la relation suivante entre les fonctions valeur :

$$V_R(b, a) = V_*(a, b). \quad (3.3)$$

En utilisant la transformation classique  $\theta_R(x, p, u) = (x \circ \phi, -p \circ \phi, u \circ \phi)$  et en notant  $(\hat{x}, \hat{p}, \hat{u}) = \theta_R(x, p, u)$ , on peut montrer facilement que  $(\text{OCP}_*)$  est équivalent à  $(\text{ROCP}_*)$  dans le sens où

$$\begin{aligned} (x, p, -1, u) \text{ est une BC-extrémale associée à une solution } (x, u) \text{ de } (\text{OCP}_*) \\ \iff (\hat{x}, \hat{p} - 1, \hat{u}) \text{ est une BC-extrémale associée à une solution } (\hat{x}, \hat{u}) \text{ de } (\text{ROCP}_*). \end{aligned}$$



Comme  $(\text{ROCP}_*)$  est de la même forme que  $(\text{OCP}_*)$  et que la fonction valeur  $V_R$  est différentiable en  $(b, a)$ , on peut appliquer l'équation (3.1) à  $(\text{ROCP}_*)$  :

$$\nabla_b V_R(\hat{x}(t_0), \hat{x}(t_f)) = -\hat{p}(t_0) = -(-p \circ \phi)(t_0) = p(t_f),$$

où  $\nabla_b$  correspond à la dérivée par rapport à la variable  $b$ , qui est ici la première variable de  $V_R$ . Finalement, en utilisant (3.3), on a directement

$$\nabla_b V_*(a, b) = \nabla_b V_R(b, a) = p(t_f). \quad \blacksquare$$

L'objectif est maintenant d'appliquer ce théorème aux problèmes  $(\text{OCP}_{i, X_i, X_{i+1}})$  pour tout  $i \in \mathbb{N}_N$ . On remarque que ces problèmes ont la même formulation que le problème  $(\text{OCP}_*)$ . Sous les hypothèses (H5) et (H6), on peut appliquer le théorème 3.1.5 sur les problèmes  $(\text{OCP}_{i, X_i, X_{i+1}})$  : si  $X$  est une solution du problème  $(\text{BOCP})$ , alors pour tout  $i \in \mathbb{N}_N$  il existe une BC-extrémale  $(\bar{x}_i, \bar{p}_i, -1, \bar{u}_i)$  associée à une solution  $(\bar{x}_i, \bar{u}_i)$  du problème  $(\text{OCP}_{i, X_i, X_{i+1}})$  et

$$\nabla V_i(X_i, X_{i+1}) = (-\bar{p}_i(t_i), \bar{p}_i(t_{i+1})).$$

Le problème  $(\text{NCBOCP})$  est donc équivalent à

$$\left\{ \begin{array}{l} \forall i \in \mathbb{N}_N, \text{ il existe une BC-extrémale } (\bar{x}_i, \bar{p}_i, -1, \bar{u}_i) \\ \text{associée à une solution } (\bar{x}_i, \bar{u}_i) \text{ de } (\text{OCP}_{i, X_i, X_{i+1}}), \\ (-\bar{p}_0(t_0), \bar{p}_N(t_{N+1})) - c'(X_0, X_{N+1})^\top \lambda = 0, \\ \bar{p}_{i-1}(t_i) - \bar{p}_i(t_i) = 0, \quad i \in \{1, \dots, N\}, \\ c(X_0, X_{N+1}) = 0, \quad X \in \mathcal{X}. \end{array} \right.$$

En remplaçant dans la première ligne « être solution » par les conditions nécessaires correspondantes, et en utilisant la notation  $\bar{z}_i = (\bar{x}_i, \bar{p}_i)$ , on obtient

$(\text{NCBOCP})$

$$\implies \left\{ \begin{array}{ll} \dot{\bar{z}}_i(t) = \vec{h}(t, \bar{z}_i(t), -1, \bar{u}_i(t)), & t \in \Delta_i \text{ p.p., } i \in \mathbb{N}_N, \\ h(t, \bar{z}_i(t), -1, \bar{u}_i(t)) = \max_{w \in U(t)} h(t, \bar{z}_i(t), -1, w), & t \in \Delta_i \text{ p.p., } i \in \mathbb{N}_N, \\ \bar{x}_i(t_i) - X_i = 0, & i \in \mathbb{N}_N, \\ \bar{x}_i(t_{i+1}) - X_{i+1} = 0, & i \in \mathbb{N}_N, \\ (-\bar{p}_0(t_0), \bar{p}_N(t_{N+1})) - c'(X_0, X_{N+1})^\top \lambda = 0, & \\ \bar{p}_{i-1}(t_i) - \bar{p}_i(t_i) = 0, & i \in \{1, \dots, N\}, \\ c(X_0, X_{N+1}) = 0, \quad X \in \mathcal{X}, & \end{array} \right.$$

qui peut encore s'écrire

$$\left\{ \begin{array}{ll} \dot{\bar{z}}_i(t) = \vec{h}(t, \bar{z}_i(t), -1, \bar{u}_i(t)), & t \in \Delta_i \text{ p.p., } i \in \mathbb{N}_N, \\ h(t, \bar{z}_i(t), -1, \bar{u}_i(t)) = \max_{w \in U(t)} h(t, \bar{z}_i(t), -1, w), & t \in \Delta_i \text{ p.p., } i \in \mathbb{N}_N, \\ (-\bar{p}_0(t_0), \bar{p}_N(t_{N+1})) - c'(\bar{x}_0(t_0), \bar{x}_N(t_{N+1}))^\top \lambda = 0, \\ \bar{x}_{i-1}(t_i) - \bar{x}_i(t_i) = 0, & i \in \{1, \dots, N\}, \\ \bar{p}_{i-1}(t_i) - \bar{p}_i(t_i) = 0, & i \in \{1, \dots, N\}, \\ c(\bar{x}_0(t_0), \bar{x}_N(t_{N+1})) = 0. \end{array} \right.$$

ou encore, en posant pour tout  $t \in [t_0, t_f]$

$$x(t) = \bar{x}_i(t), \quad p(t) = \bar{p}_i(t), \quad u(t) = \bar{u}_i(t) \quad \text{quand } t \in \Delta_i,$$

et en notant  $z = (x, p)$ ,

$$\left\{ \begin{array}{ll} \dot{z}(t) = \vec{h}(t, z(t), -1, u(t)), & t \in [t_0, t_f] \text{ p.p.,} \\ h(t, z(t), -1, u(t)) = \max_{w \in U(t)} h(t, z(t), -1, w), & t \in [t_0, t_f] \text{ p.p.,} \\ (-p(t_0), p(t_f)) - c'(x(t_0), x(t_f))^\top \lambda = 0, \\ c(x(t_0), x(t_{N+1})) = 0, \end{array} \right.$$

$$\iff \left\{ \begin{array}{ll} \dot{z}(t) = \vec{h}(t, z(t), -1, u(t)), & t \in [t_0, t_f] \text{ p.p.,} \\ h(t, z(t), -1, u(t)) = \max_{w \in U(t)} h(t, z(t), -1, w), & t \in [t_0, t_f] \text{ p.p.,} \\ g(z(t_0), z(t_f)) = 0, \end{array} \right.$$

$\iff$  (NCOCP)

Enfin, nous venons de montrer la proposition suivante

**Proposition 3.1.6 – Lien entre les conditions nécessaires d'optimalité**

*Sous les hypothèses (H5), (H6) et (H7), si le couple  $(X, \lambda)$  est une solution du problème (NCBOCP), alors il existe une BC-extrémale  $(x, p, -1, u)$  associée au problème (OCP), ou autrement dit une solution de (NCOCP), tels que*

$$\forall i \in \mathbb{N}_{N+1}, \quad X_i = x(t_i).$$

La réciproque de cette proposition n'est pas montrée. En effet, si nous avons une solution  $(z, -1, u)$  de (NCOCP), il est possible de montrer facilement que  $\forall i \in \mathbb{N}_N$  les points  $(z|_{\Delta_i}, -1, u|_{\Delta_i})$  sont des BC-extrémales des problèmes  $(\text{OCP}_{i, x(t_i), x(t_{i+1})})$ . Nous ne pouvons cependant pas affirmer aussi simplement que le couple  $(x|_{\Delta_i}, u|_{\Delta_i})$

est une solution de  $(\text{OCP}_{i,x(t_i),x(t_{i+1})})$  car le principe du maximum ne donne pas des conditions suffisantes d'optimalité dans le cas général. Il serait cependant possible de montrer la réciproque de cette proposition si on suppose l'existence de solution des problèmes  $(\text{OCP}_{i,x(t_i),x(t_{i+1})})$  et qu'il existe une unique BC-extrémale normale (avec  $p^0 = -1$ ). À ce moment-là, les conditions d'optimalité du principe du maximum deviennent des conditions nécessaires et suffisantes.

### 3.1.5 Graphe commutatif

Nous avons créé en section 3.1.1 le problème (BOCP) qui correspond à une formulation à deux niveaux du problème de commande optimale (OCP). Nous avons montré dans la section 3.1.2 le lien entre les solutions des deux formulations, à travers les propositions 3.1.3-3.1.4. Nous avons ensuite écrit dans la section 3.1.3 des conditions nécessaires d'optimalité (NCBOCP) de (BOCP) et nous avons montré dans la proposition 3.1.6 de la section 3.1.4 leur lien avec (NCOCP). Ces résultats sont résumés dans le graphe commutatif de la figure 3.1.

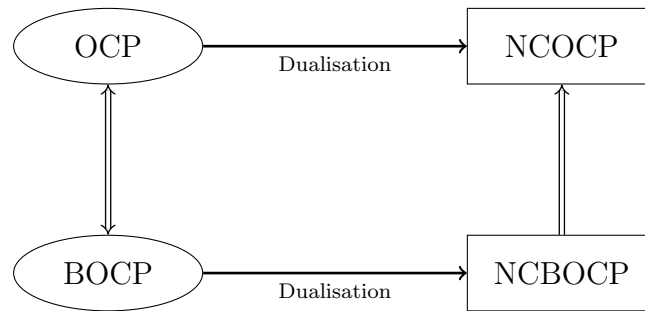


FIGURE 3.1 – Nouveau graphe commutatif entre (OCP) et (NCOCP).

## 3.2 Méthode numérique à deux niveaux

Dans les sections 2.3 et 3.1, nous avons introduit deux nouveaux points de vue sur la méthode de tir multiple : le premier en parallélisant et le second en décomposant à deux niveaux le problème de commande optimale (OCP). Ces résultats peuvent être synthétisés par le graphe commutatif suivant

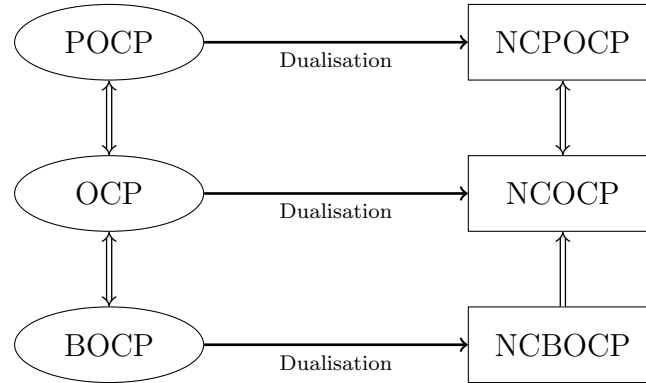


FIGURE 3.2 – Graphe commutatif entre (OCP) et (NCOCP), à travers le point de vue à deux niveaux et celui de la parallélisation.

Numériquement, nous avons vu qu'appliquer la méthode de tir simple sur (POCP) mène directement à la méthode de tir multiple déjà bien connue. A contrario, nous allons voir que le point de vue à deux niveaux permet de développer une nouvelle méthode numérique, à savoir la méthode Macro-Micro.

### 3.2.1 Idée principale

Supposons dans cette section que pour tout  $i \in \mathbb{N}_N$ , les fonctions valeur  $V_i$  sont connues et différentiables sur le domaine  $\mathring{\Omega}_i$ . Dans ce cas, la résolution du problème (OCP) peut se décomposer en deux étapes :

- dans la première étape, il s'agit de trouver un vecteur des états intermédiaires  $X^*$  solution du problème d'optimisation (BOCP) ;
- la seconde étape consiste à résoudre les  $N + 1$  problèmes  $(\text{OCP}_{i, X_i^*, X_{i+1}^*})$  afin d'obtenir les solutions  $(\bar{x}_i, \bar{u}_i)$  sur  $\Delta_i$ , pour tout  $i \in \mathbb{N}_N$ .

Tout d'abord, on remarque que dans la seconde étape, les problèmes de commande optimale sont indépendants et peuvent donc être résolus de manière parallèle par la méthode de tir simple décrite dans la section 2.2.1. De plus, le théorème 3.1.5 fournit directement pour tout  $i \in \mathbb{N}_N$  un zéro de la fonction de tir simple associée au problème  $(\text{OCP}_{i, X_i^*, X_{i+1}^*})$ , à savoir le vecteur  $z_i^* = (X_i^*, -\nabla_a V_i(X_i^*, X_{i+1}^*))$ . Une BC-extrémale normale  $(z, -1, u)$  du problème (OCP) est alors obtenue en récupérant les  $N + 1$  trajectoires d'état, de co-état et de commande  $(z_i(\cdot), -1, u_i(\cdot))$  issues de l'appel à  $\exp_{\vec{h}}(t_{i+1}, t_i, z_i^*, -1)$  pour tout  $i \in \mathbb{N}_N$ .

Finalement, si les fonctions valeur sont connues et différentiables, la méthode de résolution du problème de commande optimale (OCP) est réduite à la résolution d'un problème d'optimisation (BOCP) et de  $N + 1$  appels à l'application exponentielle sur les intervalles de temps  $\Delta_i$  qui peuvent être parallélisés.

### 3.2.2 Les problèmes (Macro) et (Micro)

La connaissance des fonctions valeur  $V_i$ , pour tout  $i \in \mathbb{N}_N$ , est une hypothèse forte, et non réaliste dans beaucoup d'applications. L'évaluation de  $V_i$  en un point

$(X_i, X_{i+1}) \in \mathring{\Omega}_i$  résulte dans la plupart des cas de la résolution numérique du problème de commande optimale ( $\text{OCP}_{i, X_i, X_{i+1}}$ ), qui est généralement assez coûteuse. Notre idée est de remplacer les fonctions valeur  $V_i$  par des approximations  $C_i$  pour tout  $i \in \mathbb{N}_N$ . On propose alors de résoudre d'abord

$$(\text{Macro}) \quad \begin{cases} \min_X C(X) = \sum_{i=0}^N C_i(X_i, X_{i+1}), \\ \text{s.c. } X \in \mathring{\mathcal{X}}, \quad c(X_0, X_{N+1}) = 0, \end{cases}$$

pour obtenir le vecteur d'états intermédiaires « optimal »  $\hat{X} = (\hat{X}_0, \dots, \hat{X}_{N+1})$ , et ensuite de résoudre les  $N + 1$  problèmes de commande optimale indépendants

$$(\text{Micro}) \quad \begin{cases} \min_{x,u} \int_{t_i}^{t_{i+1}} f^0(t, x(t), u(t)) dt, \\ \text{s.c. } \dot{x}(t) = f(t, x(t), u(t)), & t \in \Delta_i \text{ p.p.}, \\ u(t) \in U(t), & t \in \Delta_i, \\ x(t_i) = \hat{X}_i, \quad x(t_{i+1}) = \hat{X}_{i+1}, \end{cases}$$

Il s'agit d'une méthode hiérarchique, que nous appellerons *Macro-Micro*. Un schéma illustratif de cette méthode est donné dans la figure 3.3.

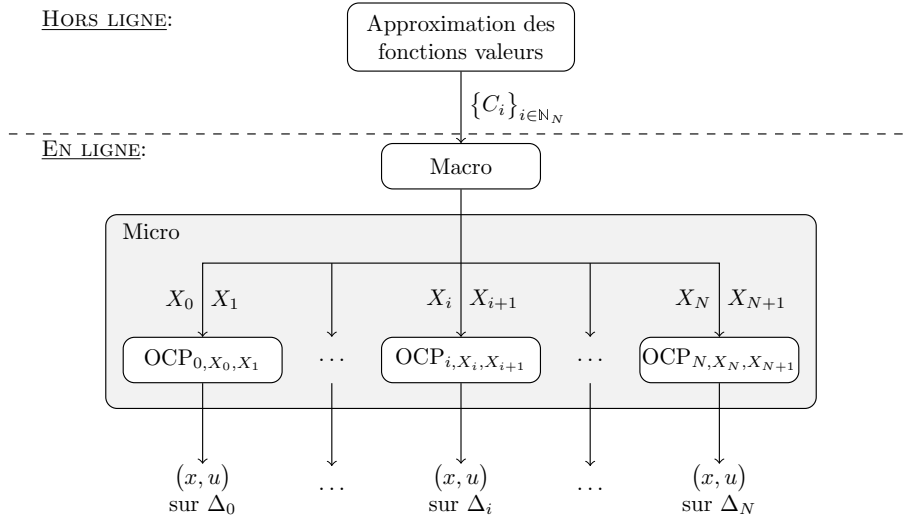


FIGURE 3.3 – Schéma de la méthode Macro-Micro.

Les approximations des fonctions valeur seront modélisées par des réseaux de neurones dans le chapitre 5. Comme nous pouvons le constater, le problème (*Micro*) est composé de l'ensemble des problèmes de commande optimale ( $\text{OCP}_{i, \hat{X}_i, \hat{X}_{i+1}}$ ), pour tout  $i \in \mathbb{N}_N$ . En comparaison avec la méthode présentée dans la section 3.2.1, le vecteur

$$(\hat{X}_i, -\nabla_a C_i(\hat{X}_i, \hat{X}_{i+1})) \quad (3.4)$$

n'est plus nécessairement un zéro de la fonction de tir simple associée, et donc un solveur de type Newton doit être utilisé. Ce vecteur nous fournit cependant une initialisation

naturelle - généralement difficile à estimer - pour la méthode de tir simple. L'idée d'utiliser le théorème 3.1.5 pour initialiser la méthode de tir a déjà été proposée dans [24], où l'approximation de la fonction valeur est obtenue avec la méthode de programmation dynamique sur une grille grossière. Une méthode similaire à la méthode Macro-Micro est par ailleurs développée dans [44] pour un problème de commande optimale stochastique en sélectionnant la bonne approximation  $C_i$  parmi quatre fonctions préalablement calculées en utilisant la méthode de programmation dynamique.

### 3.2.3 Structure du problème (Macro)

Le problème (Macro) est un problème d'optimisation qui possède une certaine structure. En effet, on peut construire le lagrangien  $L_\lambda : (\mathbb{R}^n)^{N+2} \rightarrow \mathbb{R}$  associé au problème (Macro), paramétré par  $\lambda \in \mathbb{R}^p$ , de la façon suivante

$$L_\lambda(X) = \sum_{i=0}^N C_i(X_i, X_{i+1}) - c_\lambda(X_0, X_{N+1}),$$

où  $c_\lambda(X_0, X_{N+1}) = (\lambda \mid c(X_0, X_{N+1}))$ . Le gradient du lagrangien  $L_\lambda$  au point  $X \in \mathring{\mathcal{X}}$  est donné par

$$\begin{aligned} \nabla L_\lambda(X) &= \left[ \nabla_a C_0(X_0, X_1) - \nabla_a c_\lambda(X_0, X_{N+1}) \right] e_0 \\ &\quad + \sum_{i=1}^N \left[ \nabla_b C_{i-1}(X_{i-1}, X_i) + \nabla_a C_i(X_i, X_{i+1}) \right] e_i \\ &\quad + \left[ \nabla_b C_N(X_N, X_{N+1}) - \nabla_b c_\lambda(X_0, X_{N+1}) \right] e_{N+1}, \end{aligned}$$

où pour tout  $i \in \mathbb{N}_{N+1}$ , le vecteur  $e_i$  est le vecteur de  $(\mathbb{R}^n)^{N+2}$  suivant

$$e_i = \left( \overbrace{0_{\mathbb{R}^n}, \dots, 0_{\mathbb{R}^n}}^{i \text{ fois}}, \quad 1_{\mathbb{R}^n}, \quad \overbrace{0_{\mathbb{R}^n}, \dots, 0_{\mathbb{R}^n}}^{N+1-i \text{ fois}} \right).$$

De plus, en supposant que les fonctions  $C_i$  sont de classe  $\mathcal{C}^2$ , la matrice hessienne  $\nabla^2 L_\lambda(X)$  de  $L_\lambda$  au point  $X \in \mathring{\mathcal{X}}$  est donnée par

$$\begin{aligned} \nabla^2 L_\lambda(X) &= (\nabla_{a,a}^2 C_0(X_0, X_1) - \nabla_{a,a}^2 c_\lambda(X_0, X_{N+1})) E_{0,0} \\ &\quad + \nabla_{b,a}^2 C_0(X_0, X_1) E_{0,1} - \nabla_{b,a}^2 c_\lambda(X_0, X_{N+1}) E_{0,N+1} \\ &\quad + \sum_{i=1}^N \left[ \nabla_{a,b}^2 C_{i-1}(X_{i-1}, X_i) E_{i,i-1} + \nabla_{b,a}^2 C_i(X_i, X_{i+1}) E_{i,i+1} \right. \\ &\quad \left. + (\nabla_{b,b}^2 C_{i-1}(X_{i-1}, X_i) + \nabla_{a,a}^2 C_i(X_i, X_{i+1})) E_{i,i} \right] \\ &\quad - \nabla_{a,b}^2 c_\lambda(X_0, X_{N+1}) E_{N+1,0} + \nabla_{a,b}^2 C_N(X_N, X_{N+1}) E_{N+1,N} \\ &\quad + (\nabla_{b,b}^2 C_N(X_N, X_{N+1}) - \nabla_{b,b}^2 c_\lambda(X_0, X_{N+1})) E_{N+1,N+1}, \end{aligned}$$

où la matrice  $E_{i,j} \in \mathcal{M}_{n(N+2),n(N+2)}(\mathbb{R})$  est définie pour tout  $i \in \mathbb{N}_{N+1}$  et pour tout  $j \in \mathbb{N}_{N+1}$  par

$$E_{i,j} = \left[ \begin{array}{cc} \overbrace{\begin{matrix} 0_n & \cdots & 0_n \\ \vdots & & \vdots \\ 0_n & \cdots & 0_n \end{matrix}}^{j \text{ fois}} & \overbrace{\begin{matrix} 0_n & 0_n & \cdots & 0_n \\ \vdots & \vdots & & \vdots \\ 0_n & 0_n & \cdots & 0_n \end{matrix}}^{N+1-j \text{ fois}} \\ \left. \begin{array}{cc} 0_n & 0_n \\ 0_n & 1_n \\ 0_n & 0_n \\ \vdots & \vdots \\ 0_n & 0_n \end{array} \right\} \begin{array}{l} i \text{ fois} \\ \\ \\ \\ \end{array} & \left. \begin{array}{cc} 0_n & 0_n \\ 0_n & 0_n \\ 0_n & 0_n \\ \vdots & \vdots \\ 0_n & 0_n \end{array} \right\} \begin{array}{l} \\ \\ \\ N+1-i \text{ fois} \\ \end{array} \end{array} \right]$$

où  $0_n$  et  $1_n$  sont les matrices de  $\mathcal{M}_{n,n}(\mathbb{R})$  respectivement remplies de 0 et de 1. Nous venons d'établir que la hessienne de  $L_\lambda$  peut être exprimée simplement à partir des matrices hessiennes des fonctions  $C_i$  et de la fonction  $c_\lambda$ . Pour certains types d'approximation choisis pour les fonctions  $C_i$ , la matrice  $\nabla^2 L_\lambda$  peut donc être calculée de manière peu coûteuse en termes de nombre de calculs et de place mémoire, ce qui rend possible l'utilisation de méthodes d'optimisation numérique d'ordre 2 pour la résolution du problème (Macro).

De plus, malgré son apparente complexité, cette matrice hessienne possède une structure remarquable (quasiment tridiagonale par blocs) et très creuse. En effet, si on note par  $*$  des blocs non nuls de  $\mathcal{M}_{n,n}(\mathbb{R})$ , alors  $\nabla^2 L_\lambda(X)$  est de la forme suivante

$$\nabla^2 L_\lambda(X) = \begin{bmatrix} * & * & 0 & \cdots & 0 & * \\ * & * & * & \cdots & \vdots & 0 \\ 0 & * & \cdots & \cdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & \vdots & \vdots & \vdots & * & * \\ * & 0 & \cdots & 0 & * & * \end{bmatrix}.$$

La structure spécifique de cette matrice pourrait être exploitée pour accélérer encore la résolution de (Macro).

**Remarque.** Nous n'avons pas cherché à exploiter ces caractéristiques de la matrice hessienne du lagrangien dans cette thèse. Les améliorations qui pourraient en résulter, sûrement centrales pour permettre l'embarquabilité de la méthode Macro-Micro, restent cependant une perspective très intéressante dans le cadre d'une application industrielle.

### 3.2.4 Analyse de l'erreur

Pour  $N$  fixé, le vecteur  $\hat{X} = (\hat{X}_0, \dots, \hat{X}_{N+1}) \in (\mathbb{R}^n)^{N+2}$ , solution du problème (Macro), n'a aucune raison d'être solution de (BOCP) puisque les fonctions  $C_i$  sont des approximations des fonctions valeur  $V_i$ , pour tout  $i \in \mathbb{N}_N$ . La méthode Macro-Micro est

donc une méthode sous-optimale. Il est cependant possible d'évaluer l'erreur de cette méthode, c'est-à-dire la distance en coût entre la solution trouvée par cette nouvelle méthode et la solution optimale.

En notant  $X^* = (X_0^*, \dots, X_{N+1}^*)$  une solution du problème (BOCP), le coût optimal correspond à  $V(X^*)$ . Par ailleurs, la méthode Macro-Micro fournit les fonctions valeur  $V_i(\hat{X}_i, \hat{X}_{i+1})$  par la résolution du problème (Micro). Le coût de la solution Macro-Micro s'écrit donc

$$\sum_{i=0}^N V_i(\hat{X}_i, \hat{X}_{i+1}) = V(\hat{X}).$$

Il est possible de relier un objectif de précision  $e$  de la fonction valeur à l'optimum

$$|V(\hat{X}) - V(X^*)| \leq e$$

aux erreurs d'approximation  $\varepsilon_i$  des fonctions valeur sur chaque  $\Delta_i$

$$\forall (a, b) \in \mathring{\Omega}_i, \quad |V_i(a, b) - C_i(a, b)| \leq \varepsilon_i$$

via la proposition suivante.

**Proposition 3.2.1 – Erreur de la méthode Macro-Micro**

Étant donné  $e > 0$ , si  $\max_{i \in \mathbb{N}_N} \varepsilon_i \leq \frac{e}{2(N+1)}$  alors  $|V(\hat{X}) - V(X^*)| \leq e$ .

► En utilisant l'inégalité triangulaire, on a directement pour tout  $X \in \mathring{\mathcal{X}}$

$$|V(X) - C(X)| \leq \sum_{i=0}^N \varepsilon_i \leq (N+1) \max_{i \in \mathbb{N}_N} \varepsilon_i \leq \frac{e}{2},$$

ou de manière équivalente

$$V(X) - \frac{e}{2} \leq C(X) \leq V(X) + \frac{e}{2}. \quad (3.5)$$

En utilisant le fait que  $X^*$  est une solution du problème (BOCP) et les inégalités (3.5) en  $X = \hat{X}$ , on obtient

$$V(X^*) - \frac{e}{2} \leq V(\hat{X}) - \frac{e}{2} \leq C(\hat{X}) \leq V(\hat{X}) + \frac{e}{2} \implies V(X^*) - V(\hat{X}) \leq e.$$

De manière analogue, en utilisant le fait que  $\hat{X}$  est une solution du problème (Macro), et les inégalités (3.5) en  $X = \hat{X}$  et en  $X = X^*$ , on obtient

$$V(\hat{X}) - \frac{e}{2} \leq C(\hat{X}) \leq C(X^*) \leq V(X^*) + \frac{e}{2} \implies V(\hat{X}) - V(X^*) \leq e.$$

Finalement, nous venons de montrer que  $|V(\hat{X}) - V(X^*)| \leq e$ . ■

L'interprétation de cette proposition est intéressante d'un point de vue industriel : pour  $N$  fixé, on peut s'assurer que la différence en coût entre la solution optimale et la solution



de la méthode Macro-Micro soit inférieure à un seuil  $e \geq 0$  en faisant en sorte que pour tout  $i \in \mathbb{N}_N$ , l'erreur maximum de nos approximations,  $\max_{i \in \mathbb{N}_N} \varepsilon_i$ , soit inférieure à  $\frac{e}{2(N+1)}$ .

### 3.2.5 Avantages et inconvénients

Dans cette partie, on considère que le coût de résolution, en termes de temps ou de nombre de calculs, du problème d'optimisation (**Macro**) est négligeable comparé au coût de résolution d'un problème de commande optimale (**OCP** $_{i, X_i, X_{i+1}}$ ), pour  $i \in \mathbb{N}_N$  et  $(X_i, X_{i+1}) \in \mathring{\Omega}_i$ . On considère donc que le coût de la méthode Macro-Micro est composé uniquement du coût des problèmes (**Micro**). Pour simplifier la discussion, on considère également que les temps intermédiaires discrétisent uniformément l'intervalle de temps initial  $[t_0, t_f]$ , c'est-à-dire que pour tout  $i \in \mathbb{N}_{N+1}$

$$t_i = t_0 + i \frac{(t_f - t_0)}{N + 1}.$$

**Temps de calcul.** Pour tout  $i \in \mathbb{N}_N$ , l'intervalle de temps  $\Delta_i$  est  $N + 1$  fois plus court que l'intervalle de temps  $[t_0, t_f]$ , ce qui implique que le calcul de  $\exp_{\vec{h}}(t_{i+1}, t_i, \cdot, \cdot)$  est  $(N + 1)$  fois plus rapide que le calcul de  $\exp_{\vec{h}}(t_f, t_0, \cdot, \cdot)$ .

Si l'on considère que le nombre d'itérations du solveur de type Newton utilisé pour la recherche de zéro d'une fonction de tir définie sur l'intervalle de temps  $[t_0, t_f]$  est égal au nombre d'itérations pour une fonction de tir définie sur  $\Delta_i$ , la résolution du problème (**OCP** $_{i, a, b}$ ) est donc  $(N + 1)$  fois plus rapide pour tout  $(a, b) \in \Omega_i$  que la résolution du problème (**OCP**) par la méthode de tir simple indirecte. Finalement, comme les problèmes de commande optimale du problème (**Micro**) sont indépendants, on peut les résoudre parallèlement, ce qui fait que la méthode Macro-Micro est  $(N + 1)$  fois plus rapide que la méthode de tir simple indirecte.

Dans la méthode de tir multiple, il est possible de paralléliser l'appel aux applications exponentielles. Cet aspect numérique est mis en avant avec le point de vue sur la parallélisation. On peut donc dire que le temps de calcul des méthodes Macro-Micro et tir multiple (parallélisée) sont du même ordre.

**Robustesse.** La méthode de tir multiple est plus robuste que la méthode de tir simple [4] car l'application exponentielle est appelée sur des intervalles de temps plus courts. En utilisant le même argument, la méthode Macro-Micro est au moins aussi robuste que la méthode de tir multiple. De plus, l'initialisation donnée par l'équation (3.4) peut permettre d'améliorer la convergence de notre solveur, et donc d'augmenter la robustesse de la méthode Macro-Micro, tout en réduisant le nombre d'itérations du solveur (et donc également le temps et le nombre de calculs).

**Méthode embarquée.** Dans une solution embarquée, la méthode de résolution doit être rapide et la moins coûteuse possible en termes de nombre de calculs pour ne pas avoir à surdimensionner les calculateurs. Dans les méthodes indirectes classiques, il faut calculer la commande sur tout l'intervalle de temps  $[t_0, t_f]$  pour la connaître sur  $\Delta_0$ . La

parallélisation de l'appel à l'application exponentielle rendue possible par la méthode de tir multiple permet de calculer  $N+1$  fois plus vite la commande optimale à condition que le calculateur embarqué soit suffisamment performant pour supporter ce calcul parallèle.

Dans la méthode Macro-Micro, l'indépendance des problèmes Micro rend possible le calcul de la commande optimale sur le seul intervalle  $\Delta_0$  : ce calcul est donc  $N+1$  fois plus rapide qu'un calcul de la commande sur l'intervalle de temps complet  $[t_0, t_f]$  et un calculateur standard reste suffisant. Il ne faut cependant pas voir la méthode Macro-Micro comme une méthode locale. En effet, la résolution du problème (Macro) permet de trouver les états intermédiaires « optimaux », et donc de garder une vision long terme.

## Conclusion

Une nouvelle méthode de contrôle optimal, la méthode Macro-Micro, a été développée dans ce chapitre. Cette méthode est basée sur une formulation à deux niveaux du problème (OCP) : le problème (BOCP). La relation entre ces deux problèmes, ainsi que la relation entre leurs conditions nécessaires, c'est-à-dire la relation entre (NCOCP) et (NCBOCP), sont données à travers la figure 3.1.

L'idée principale de cette méthode est d'approximer les fonctions valeur  $V_i$  du problème (BOCP) par les fonctions  $C_i$ , ce qui mène à une résolution hiérarchique des problèmes (Macro) et (Micro). Nous avons vu que le problème (Macro) possède une certaine structure, qui pourrait être exploitée numériquement afin de permettre une méthode de résolution du problème (Macro) efficace. Cette méthode est certes sous-optimale, mais nous avons vu avec la proposition 3.2.1 que cette sous-optimalité peut être contrôlée par les erreurs d'approximation des  $C_i$ . Finalement, les principaux avantages (rapidité, robustesse et embarquabilité) et inconvénients (sous-optimalité, construction des  $C_i$ ) de la méthode Macro-Micro ont été décrits dans la section 3.2.5.

La méthode Macro-Micro est appliquée dans le chapitre 5 au problème de répartition de couple dans les véhicules hybride introduit précédemment dans le chapitre 1. Nous constaterons que cette méthode permet d'obtenir une solution proche de la solution optimale, tout en accélérant grandement sa résolution et en réduisant le nombre de calculs. De plus, l'initialisation naturelle proposée par l'équation (3.4) permettra au solveur de converger plus rapidement.



## Préconditionnement de la fonction de tir

---

<b>4.1 Formulation augmentée</b> .....	<b>66</b>
4.1.1 Système augmenté .....	66
4.1.2 Conditions nécessaires d'optimalité .....	66
4.1.3 Ensemble accessible .....	67
<b>4.2 Interprétation géométrique du co-état</b> .....	<b>69</b>
4.2.1 Définitions et propriétés .....	69
4.2.2 Origine du co-état et cône de Boltyanski .....	72
4.2.3 Lien avec le cône tangent et le cône normal .....	73
4.2.4 Bilan .....	75
<b>4.3 Préconditionnement géométrique</b> .....	<b>76</b>
4.3.1 Principe .....	76
4.3.2 Cas 1 : l'ensemble accessible est difféomorphe à la boule .....	79
4.3.3 Cas 2 : extension au cas non difféomorphe .....	83
4.3.4 Cas général : utilisation de l'application exponentielle .....	89

Dans le chapitre précédent, nous avons proposé une méthode de résolution du problème (OCP) qui mène à la résolution des problèmes de commande optimale du problème (Micro). Ces derniers sont définis sur des intervalles de temps plus courts et ils ont la même formulation, à savoir :

$$(\text{OCP}_*) \quad \left\{ \begin{array}{l} \min_{x,u} \int_{t_0}^{t_f} f^0(t, x(t), u(t)) \, dt, \\ \text{s.c. } \dot{x}(t) = f(t, x(t), u(t)) \quad t \in [t_0, t_f] \text{ p.p.}, \\ \quad \quad u(t) \in U(t), \quad t \in [t_0, t_f], \\ \quad \quad x(t_0) = x_0, \quad x(t_f) = x_f. \end{array} \right.$$

La méthode de tir simple peut être utilisée pour résoudre ce problème de commande optimale. Nous avons vu au chapitre 2 que cette méthode est connue pour être particulièrement sensible à l'initialisation du solveur. Suite aux travaux de [23], l'objectif de ce chapitre est de proposer une méthode de preconditionnement géométrique de la fonction de tir, basée sur une interprétation géométrique du co-état et sur la transformée de Mathieu, dont l'objectif est de réduire cette sensibilité.

## 4.1 Formulation augmentée

L'objectif principal de cette section est de construire l'ensemble accessible du système augmenté. Dans ce but, nous transformons le problème (OCP<sub>\*</sub>) en son équivalent sous forme augmenté (AOCP<sub>\*</sub>) dans la section (4.1.1). Nous donnons ensuite les conditions nécessaires d'optimalité de cette nouvelle formulation dans la section 4.1.2 pour enfin, en section 4.1.3, construire l'ensemble accessible du système augmenté.

### 4.1.1 Système augmenté

On considère l'état augmenté  $\hat{x} = (x^0, x)$  où  $x^0$  correspond au coût et  $x$  à l'état. Cette notation est utilisée de la même manière si  $\hat{x}$  est une fonction absolument continue à valeur dans  $\mathbb{R}^{n+1}$  ou si  $\hat{x}$  est un vecteur de  $\mathbb{R}^{n+1}$ . Le système augmenté  $\hat{f}: \mathbb{R} \times \mathbb{R}^{n+1} \times \mathbb{R}^m$  est défini par

$$\hat{f}(t, \hat{x}, u) = (f^0(t, x, u), f(t, x, u)),$$

et le problème de commande optimale sous forme augmentée s'écrit classiquement de la manière suivante

$$(AOCP_*) \quad \begin{cases} \min_{\hat{x}, u} x^0(t_f) \\ \text{s.c. } \dot{\hat{x}}(t) = \hat{f}(t, \hat{x}(t), u(t)), & t \in [t_0, t_f] \text{ p.p.}, \\ u(t) \in U(t), & t \in [t_0, t_f], \\ \hat{x}(t_0) = \hat{x}_0 := (0, x_0), \quad x(t_f) = x_f. \end{cases}$$

Cette transformation d'un problème de commande optimale sous forme de Lagrange en forme augmentée est classique, et bien connue de la littérature [16].

### 4.1.2 Conditions nécessaires d'optimalité

Le principe du maximum présenté au chapitre 2 fournit également des conditions nécessaires d'optimalité pour un problème de commande optimale sous la forme (AOCP<sub>\*</sub>). Ainsi, si  $(\hat{x}, u)$  est solution de (AOCP<sub>\*</sub>), alors il existe  $\hat{p} = (p^0, p)$  non trivial dans  $AC([t_0, t_f], \mathbb{R}^{n+1})$  avec  $p^0(t_f) \leq 0$  tel que pour presque tout  $t \in [t_0, t_f]$  la dynamique pseudo-hamiltonienne est satisfaite

$$\begin{aligned} \dot{\hat{x}}(t) &= \nabla_{\hat{p}} \hat{h}(t, \hat{x}(t), \hat{p}(t), u(t)), \\ \dot{\hat{p}}(t) &= -\nabla_{\hat{x}} \hat{h}(t, \hat{x}(t), \hat{p}(t), u(t)), \end{aligned}$$

tout comme la condition de maximisation

$$\hat{h}(t, \hat{x}(t), \hat{p}(t), u(t)) = \max_{w \in U(t)} \hat{h}(t, \hat{x}(t), \hat{p}(t), w),$$

où  $\hat{h}(t, \hat{x}, \hat{p}, u) = (\hat{p} | \hat{f}(t, \hat{x}, u))$  est le pseudo-hamiltonien du système augmenté.

Tout comme nous l'avons fait précédemment et pour avoir un système de notation simple et compréhensible, on propose de noter  $\hat{z} = (\hat{x}, \hat{p})$  le couple état/co-état du

système augmenté. De plus, le co-état du système augmenté  $\hat{p} = (p^0, p)$  est composé du multiplicateur  $p^0$  associé au coût et du multiplicateur  $p$  associé à l'état (que l'on appelle co-état). On note également  $z = (x, p)$  le couple état/co-état. Finalement, ces notations peuvent être utilisées pour des fonctions absolument continues ou pour des vecteurs.

En définissant le champ de vecteurs pseudo-hamiltonien du système augmenté par

$$\vec{h}(t, \hat{z}, u) = \mathbf{J}_{2(n+1)} \nabla_{\hat{z}} \hat{h}(t, \hat{z}, u),$$

où  $\mathbf{J}_{2(n+1)}$  est la matrice carrée symplectique de dimension  $2(n+1)$ , définie précédemment par l'équation (2.8), les conditions nécessaires d'optimalité du problème (AOCP<sub>\*</sub>) peuvent s'exprimer sous la forme suivante

$$\begin{cases} \dot{\hat{z}}(t) = \vec{h}(t, \hat{z}(t), u(t)), & t \in [t_0, t_f] \text{ p.p.}, \\ \hat{h}(t, \hat{z}(t), u(t)) = \max_{w \in \mathcal{U}(t)} \hat{h}(t, \hat{z}(t), w), & t \in [t_0, t_f] \text{ p.p.}, \\ \hat{x}(t_0) = \hat{x}_0, \quad x(t_f) = x_f, \\ p^0(t_f) \leq 0, \quad \hat{p} \neq 0. \end{cases} \quad (4.1)$$

**Remarque 4.1.1.** Comme la dynamique d'état augmentée ne dépend pas de  $x^0$ , le multiplicateur associé au coût  $p^0(\cdot)$  est constant sur  $[t_0, t_f]$ , et on propose de le caractériser par sa condition finale  $p^0(\cdot) = p^0(t_f)$ . Cette valeur correspond directement au multiplicateur  $p^0$  dans la formulation classique. Cependant, cette différence implique qu'une extrémale pour le système augmenté est un couple  $(\hat{z}, u)$  (et non un triplet) qui satisfait la dynamique pseudo-hamiltonienne ainsi que la condition de maximisation.

Sous l'hypothèse (H2) page 26, l'application exponentielle  $\exp_{\vec{h}}(\hat{p}_0)$  du champ de vecteurs pseudo-hamiltonien du système augmenté est bien définie, c'est la solution au temps final  $t_f$  du problème

$$\begin{cases} \dot{\hat{z}}(t) = \vec{h}(t, \hat{z}(t), u(t)), & t \in [t_0, t_f] \text{ p.p.}, \\ \hat{h}(t, \hat{z}(t), u(t)) = \max_{w \in \mathcal{U}(t)} \hat{h}(t, \hat{z}(t), w), & t \in [t_0, t_f] \text{ p.p.}, \\ \hat{z}(t_0) = (\hat{x}_0, \hat{p}_0). \end{cases}$$

### 4.1.3 Ensemble accessible

L'intérêt principal de la formulation augmentée est de permettre la construction de l'ensemble accessible augmenté  $\mathcal{A}$ , qui correspond à l'ensemble des états augmentés accessibles au temps  $t_f$  à partir de la condition initiale  $\hat{x}_0 := (0, x_0)$  au temps  $t_0$ . Cet ensemble correspond directement à l'ensemble accessible défini dans la section 3.1.1 du système augmenté. On définit donc  $\hat{x}(u)$  la solution au temps  $t_f$  du problème de Cauchy suivant

$$\begin{cases} \dot{\hat{x}}(t) = \hat{f}(t, \hat{x}(t), u(t)), & t \in [t_0, t_f] \text{ p.p.}, \\ \hat{x}(t_0) = \hat{x}_0. \end{cases}$$

On pose  $\mathcal{U}$  l'ensemble des *contrôles admissibles*, c'est-à-dire l'ensemble des contrôles  $u \in L^\infty([t_0, t_f], \mathbb{R}^m)$  tel que pour tout temps  $t \in [t_0, t_f]$ , on a  $u(t) \in U(t)$  et tel que  $\hat{x}(u)$  soit bien défini. Finalement, l'ensemble accessible augmenté  $\mathcal{A}$  est défini par

$$\mathcal{A} = \hat{x}(\mathcal{U}). \quad (4.2)$$

### Proposition 4.1.1

Si  $(\hat{x}, u)$  est solution de (AOCP<sub>\*</sub>) alors  $\hat{x}(t_f) \in \partial\mathcal{A}$ .

► Soit  $(\hat{x}_*, u_*)$  une solution de (AOCP<sub>\*</sub>). Supposons que  $\hat{x}_*(t_f) \notin \partial\mathcal{A}$ . Il existe donc un point  $(\hat{x}, u)$  admissible pour (AOCP<sub>\*</sub>) tels que

$$\hat{x}(t_0) = \hat{x}_0 \quad \text{et} \quad \hat{x}(t_f) = (x^0, x_T) \quad \text{avec} \quad x^0 < x_*^0(t_f).$$

Ce point serait donc un point admissible avec un coût plus faible, ce qui n'est pas possible car  $(\hat{x}_*, u_*)$  est une solution. Par contradiction,  $\hat{x}_*(t_f) \in \partial\mathcal{A}$ . ■

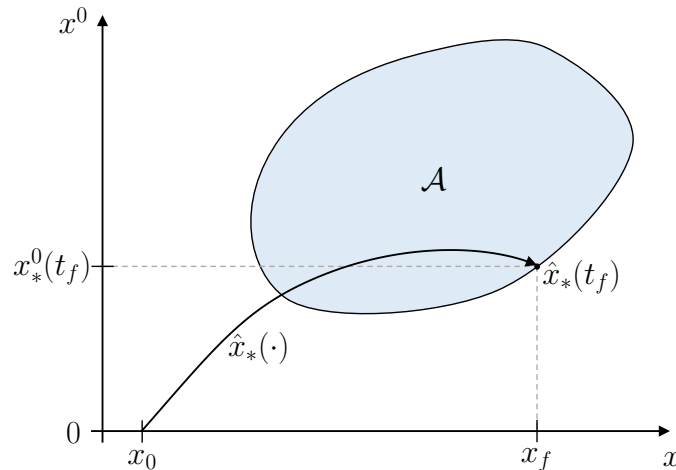


FIGURE 4.1 – Illustration de la trajectoire optimale  $\hat{x}_*(\cdot)$  et de l'ensemble accessible augmenté  $\mathcal{A}$ . Comme on peut le constater,  $\hat{x}_*(t_f) \in \partial\mathcal{A}$ .

Cette proposition, illustrée dans la figure 4.1, nous permet de déduire des propriétés sur l'ensemble accessible augmenté. Supposons qu'il existe une solution  $(\hat{x}_*, u_*)$  de (AOCP<sub>\*</sub>). L'ensemble  $\mathcal{A}$  n'est donc pas vide. De plus, l'ensemble  $\mathcal{A}$  n'est pas ouvert. En effet, si  $\mathcal{A}$  était un ouvert, alors on aurait

$$\partial\mathcal{A} \cap \mathcal{A} = \emptyset$$

et par conséquent, puisque  $\hat{x}_*(t_f) \in \partial\mathcal{A}$ , alors  $\hat{x}_*(t_f) \notin \mathcal{A}$ , ce qui est incompatible avec le fait que  $(\hat{x}_*, u_*)$  soit une solution. C'est pourquoi, pour une question de simplicité, on considère dans la suite que l'ensemble accessible augmenté  $\mathcal{A}$  est un ensemble fermé non vide.

## 4.2 Interprétation géométrique du co-état

Dans cette partie, nous allons discuter de l'origine du co-état, et plus particulièrement de sa construction. En effet, la preuve du principe du maximum est une preuve constructive qui peut être trouvée dans [1]. Nous allons uniquement la survoler ici. Commençons par introduire des définitions et des propriétés importantes.

### 4.2.1 Définitions et propriétés

#### Définition 4.2.1 – Cône

Un ensemble  $C \subset \mathbb{R}^n$  est un *cône* si et seulement si pour tout  $x \in C$  et pour tout  $\lambda > 0$ ,  $\lambda x \in C$ . De plus, un cône est dit

- *convexe* si pour tout  $x, y \in C$  et pour tout  $\alpha, \beta > 0$ ,  $\alpha x + \beta y \in C$ ;
- *pointé* si  $0 \in C$ ;
- *saillant* si  $x \in C$  et  $-x \in C$  implique que  $x = 0$ .

Un cône est dit *propre* s'il est convexe, pointé et saillant.

#### Définition 4.2.2 – Cône polaire

Le *cône polaire*  $C^\circ$  d'un cône  $C$  de  $\mathbb{R}^n$  est défini comme son dual négatif

$$C^\circ := \{y \in \mathbb{R}^n \mid (y \mid x) \leq 0, \forall x \in C\}.$$

#### Propriété 4.2.3

Si  $A$  et  $B$  sont deux cônes de  $\mathbb{R}^n$  tels que  $A \subset B$ , alors  $B^\circ \subset A^\circ$ .

► Soit  $A$  et  $B$  deux cônes de  $\mathbb{R}^n$  tels que  $A \subset B$ , et soit  $y \in B^\circ$ . Par définition, on a pour tout  $x \in B$

$$(y \mid x) \leq 0.$$

Comme  $A \subset B$ , on a donc le même résultat pour tout  $y \in A$ , donc  $x \in A^\circ$ . ■

#### Définition 4.2.4 – Cône tangent de Bouligand

Soit  $E$  un sous-ensemble non vide de  $\mathbb{R}^n$  et un point  $x \in E$ . Le *cône tangent* (de Bouligand)  $T_E(x)$  de  $E$  au point  $x$  est défini par

$$T_E(x) := \{d \in \mathbb{R}^n \mid \exists d_k \rightarrow d, \exists \varepsilon_k \downarrow 0, x + \varepsilon_k d_k \in E\},$$

ou, de manière équivalente, par

$$T_E(x) := \left\{ d \in \mathbb{R}^n \mid \exists x_k \in E, \exists \varepsilon_k \downarrow 0, \frac{x_k - x}{\varepsilon_k} \rightarrow d \right\}.$$



**Définition 4.2.5 – Cône normal**

Soit  $E$  un sous-ensemble non vide de  $\mathbb{R}^n$  et un point  $x \in E$ . Le *cône normal*  $N_E(x)$  de  $E$  au point  $x$  correspond au cône polaire du cône tangent de  $E$  au point  $x$  :  $N_E(x) := T_E^\circ(x)$ .

**Propriété 4.2.6 – Cône tangent et normal [19, 34]**

Soit  $E$  un sous-ensemble non vide de  $\mathbb{R}^n$  et  $x \in E$ .

- Si  $x \in \overset{\circ}{E}$  alors  $T_E(x) = \mathbb{R}^n$  et  $N_E(x) = \{0\}$ .
- Si  $x \in \partial E$ ,
  - Si  $\partial E$  est lisse en  $x$ , alors  $T_E(x)$  est le demi-espace délimité par le plan tangent à  $\partial E$  en  $x$  contenant localement  $E$  et  $N_E(x)$  est la demi-droite engendrée par la normale à ce plan tangent extérieure localement à l'ensemble  $E$ .
  - Si  $\partial E$  est non lisse en  $x$ ,
    - si  $E$  est localement convexe en  $x$ , alors  $T_E(x)$  est propre et  $N_E(x)$  est non trivial, c'est-à-dire non réduit au singleton  $\{0\}$ , c.f. Figure 4.3a.
    - si  $E$  n'est pas localement convexe en  $x$ , alors  $T_E(x)$  n'est pas propre et  $N_E(x) = \{0\}$ , c.f. Figure 4.3b.

**Proposition 4.2.7**

Si  $E$  est un sous-ensemble non vide de  $\mathbb{R}^n$ ,  $x$  un point de  $E$  et  $\phi$  un difféomorphisme de  $\mathbb{R}^n$ , alors

$$T_{\phi(E)}(\phi(x)) = J_\phi(x) T_E(x).$$

► Soit  $E$  un sous-ensemble de  $\mathbb{R}^n$ ,  $x \in E$  et  $\phi$  un difféomorphisme de  $\mathbb{R}^n$ . On pose  $y = \phi(x)$  et  $F = \phi(E)$ . Montrons que  $T_F(y) \subset J_\phi(x) T_E(x)$ . Autrement dit, en prenant  $d \in T_F(y)$ , on veut montrer que  $u := J_\phi(x)^{-1}d$  appartient à  $T_E(x)$ .

Comme  $d \in T_F(y)$ , il existe une suite  $y_k \in F$  et une suite réelle positive  $\varepsilon_k \downarrow 0$  telles que

$$d_k := \frac{y_k - y}{\varepsilon_k} \longrightarrow d.$$

Il existe donc une suite  $x_k \in E$  tels que  $y_k = \phi(x_k)$ , et nous avons

$$y_k - y = \varepsilon_k d_k \implies \phi(x_k) - \phi(x) = \varepsilon_k d_k. \quad (4.3)$$

En utilisant le développement de Taylor de  $\phi$  en  $x_k$  à l'ordre 1, nous obtenons également que

$$\phi(x_k) - \phi(x) = J_\phi(x)(x_k - x) + \|x_k - x\| \mathcal{E}(x_k - x) \quad (4.4)$$

avec  $\|\mathcal{E}(x_k - x)\| \rightarrow 0$ . En utilisant les équations (4.3) et (4.4), nous avons donc

$$d_k = J_\phi(x) \frac{x_k - x}{\varepsilon_k} + \frac{\|x_k - x\|}{\varepsilon_k} \mathcal{E}(x_k - x).$$

En posant  $u_k = J_\phi(x)^{-1}d_k$  et  $v_k = \frac{x_k - x}{\varepsilon_k}$ , on obtient

$$u_k = v_k + \|v_k\| \mathcal{E}'(x_k - x), \quad \text{avec} \quad \mathcal{E}'(x_k - x) = J_\phi(x)^{-1} \mathcal{E}(x_k - x).$$

Il nous reste à montrer que  $v_k \rightarrow u$ . Comme  $v_k = u_k - \|v_k\| \mathcal{E}'(x_k - x)$ , en utilisant l'inégalité triangulaire, on a

$$\|v_k\| \leq \|u_k\| + \|v_k\| \|\mathcal{E}'(x_k - x)\| \implies \|v_k\| (1 - \|\mathcal{E}'(x_k - x)\|) \leq \|u_k\|.$$

Puisque  $d_k \rightarrow d$ , on a  $u_k = J_\phi(x)^{-1}d_k \rightarrow u$ , et il existe donc  $M \in \mathbb{R}$  tel que  $\|u_k\| \leq M$ . De plus, comme par définition  $\|\mathcal{E}'(x_k - x)\| = \|J_\phi(x)^{-1} \mathcal{E}(x_k - x)\| \rightarrow 0$ , on a pour  $k$  suffisamment grand

$$\frac{1}{2} \|v_k\| \leq \|v_k\| (1 - \|\mathcal{E}'(x_k - x)\|).$$

Pour  $k$  suffisamment grand, on a donc

$$\frac{1}{2} \|v_k\| \leq \|v_k\| (1 - \|\mathcal{E}'(x_k - x)\|) \leq M \implies \|v_k\| \leq 2M,$$

ce qui implique directement que

$$v_k = u_k - \|v_k\| \mathcal{E}'(x_k - x) \rightarrow u.$$

Finalement, on vient de montrer que si  $d \in T_{\phi(E)}(\phi(x))$ , alors il existe une direction  $u \in T_E(x)$  tel que  $d = J_\phi(x) u$ , c'est-à-dire que

$$T_{\phi(E)}(\phi(x)) \subset J_\phi(x) T_E(x).$$

En utilisant ce résultat avec le difféomorphisme  $\phi^{-1}$ , nous avons directement que  $T_F(y) = J_\phi(x) T_E(x)$ . En effet, on a directement

$$\begin{aligned} T_E(x) &= T_{\phi^{-1}(F)}(\phi^{-1}(y)) \subset J_{\phi^{-1}}(y) T_F(y) = J_\phi(x)^{-1} T_F(y) \\ &\implies J_\phi(x) T_E(x) \subset T_{\phi(E)}(\phi(x)) \end{aligned}$$

Finalement, nous venons de montrer que

$$T_{\phi(E)}(\phi(x)) = J_\phi(x) T_E(x).$$

■

**Corollaire 4.2.8**

Si  $E$  est un sous-ensemble non vide de  $\mathbb{R}^n$ ,  $x$  un point de  $E$  et  $\phi$  un difféomorphisme de  $\mathbb{R}^n$ , alors

$$J_\phi(x)^\top N_{\phi(E)}(\phi(x)) = N_E(x).$$

► En utilisant la définition du cône normal, on a

$$N_{\phi(E)}(\phi(x)) = \{y \in \mathbb{R}^n \mid (y \mid z) \leq 0, \forall z \in T_{\phi(E)}(\phi(x))\}.$$

En utilisant la proposition 4.2.7, on a

$$\begin{aligned} N_{\phi(E)}(\phi(x)) &= \{y \in \mathbb{R}^n \mid (y \mid J_\phi(x)z) \leq 0, \forall z \in T_E(x)\} \\ &= \{y \in \mathbb{R}^n \mid (J_\phi(x)^\top y \mid z) \leq 0, \forall z \in T_E(x)\} \end{aligned}$$

et, en posant  $w = J_\phi(x)^\top y$ , on a directement

$$\begin{aligned} N_{\phi(E)}(\phi(x)) &= \{J_\phi(x)^{-\top} w \in \mathbb{R}^n \mid (w \mid z) \leq 0, \forall z \in T_E(x)\} \\ &= J_\phi(x)^{-\top} N_E(x). \end{aligned}$$

On a donc bien

$$J_\phi(x)^\top N_{\phi(E)}(\phi(x)) = N_E(x). \quad \blacksquare$$

**4.2.2 Origine du co-état et cône de Boltyanski**

Soit  $(\hat{x}, u)$  une solution du problème (AOCP<sub>\*</sub>), et on note  $\hat{x}_f = \hat{x}(t_f)$ . L'idée principale de la démonstration du principe du maximum de Pontryagin est de considérer un certain type de variations sur la commande optimale, qui sont les variations en aiguilles proposées par V. G. Boltyanski [7]. Ces variations, illustrées dans la figure 4.2a, génèrent une famille d'ensembles de commandes  $\{\mathcal{U}_\tau\}_{\tau \geq 0}$ , inclus dans  $\mathcal{U}$ , tels que  $\mathcal{U}_0 = \{u\}$ . Elles permettent de générer l'ensemble

$$\mathcal{V}_\tau = \hat{x}(\mathcal{U}_\tau) - \hat{x}_f.$$

Par construction, cet ensemble est inclus dans  $\mathcal{A} - \hat{x}_f$  pour tout  $\tau \geq 0$ , et on a bien évidemment  $\mathcal{V}_0 = \{0\}$ . On construit ensuite le cône

$$\mathcal{K} = \{d \in \mathbb{R}^{n+1} \mid \exists d_k \rightarrow d, \exists \varepsilon_k \downarrow 0, \varepsilon_k d_k \in \mathcal{V}_{\varepsilon_k}\},$$

qui correspond à un cône de directions tangentes, et que l'on nomme le cône de Boltyanski [60]. Une illustration de ce cône extraite de [7] est donnée dans la figure 4.2b.

Une des parties complexes de cette démonstration est de montrer que  $\mathcal{K}$  est un cône propre. Cette preuve n'est pas traitée ici, mais peut être trouvée dans [1]. Le cône de Boltyanski est une approximation locale convexe de  $\mathcal{A} - \hat{x}_f$  et possède un hyperplan support à l'origine :

$$\exists \lambda \in \mathbb{R}^{n+1}, \quad \lambda \neq 0, \quad (\lambda \mid d) \leq 0, \quad \forall d \in \mathcal{K}, \quad (4.5)$$

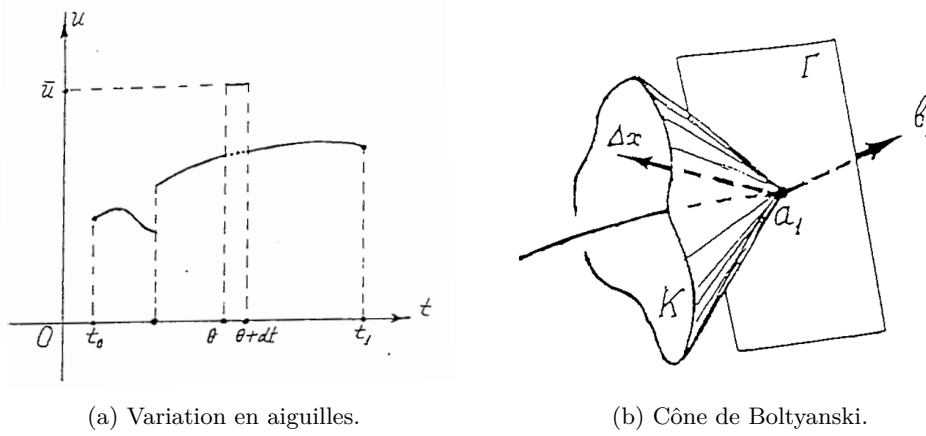


FIGURE 4.2 – Illustrations de la main de V. G. Boltyanski extraites de [7].

ou autrement dit le cône polaire du cône de Boltyanski n'est pas trivial :

$$\hat{\mathcal{K}}^\circ \neq \{0\}. \quad (4.6)$$

La fin de la preuve du principe du maximum permet d'établir que pour tout co-état augmenté final  $\hat{p}_f \in \hat{\mathcal{K}}^\circ$  non trivial tels que  $p_f^0 \leq 0$ , on peut construire une BC-extrémale associée à  $\hat{x}_f$  en intégrant en arrière la dynamique pseudo-hamiltonienne et en satisfaisant la condition de maximisation.

### 4.2.3 Lien avec le cône tangent et le cône normal

Par construction, le cône de Boltyanski est inclus dans le cône tangent

$$\mathcal{K} \subset T_{\mathcal{A}}(\hat{x}_f).$$

En effet, si  $d \in \mathcal{K}$ , alors il existe une suite de directions  $d_k \rightarrow d$  et une suite de réels décroissante  $\varepsilon_k \downarrow 0$  telles que  $\varepsilon_k d_k \in \mathcal{V}_{\varepsilon_k}$ . Comme par construction  $\mathcal{V}_{\varepsilon_k} \subset \mathcal{A} - \hat{x}_f$ , on a donc

$$\hat{x}_f + \varepsilon_k d_k \in \mathcal{A},$$

ce qui est la définition pour  $d$  d'appartenir au cône tangent  $T_{\mathcal{A}}(\hat{x}_f)$ . En utilisant la propriété 4.2.3 et la définition du cône normal, on a donc

$$N_{\mathcal{A}}(\hat{x}_f) \subset \mathcal{K}^\circ.$$

Ces inclusions sont illustrées dans la figure 4.3a.

Comme nous l'avons vu précédemment avec l'équation (4.6), le cône polaire du cône de Boltyanski est nécessairement non trivial. Cependant, comme on l'a vu dans la propriété 4.2.6, il existe certains cas où le cône normal est réduit au singleton  $\{0\}$ , comme illustré dans la figure 4.3b.

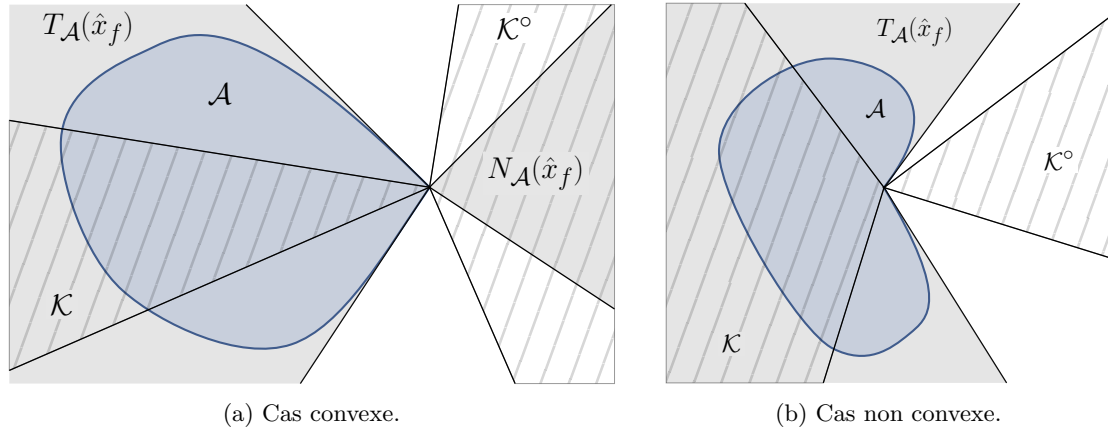


FIGURE 4.3 – Illustration d'un ensemble accessible augmenté  $\mathcal{A}$  non vide fermé, convexe (à gauche) et non convexe (à droite). Le cône  $\mathcal{K}$  est un cône propre inclus dans le cône tangent, et correspond à une illustration du cône de Boltyanski.

Finalement, nous venons de montrer le théorème suivant

#### Théorème 4.2.9

Si  $(\hat{x}, \hat{u})$  est une solution de (AOCP<sub>\*</sub>) et  $N_{\mathcal{A}}(\hat{x}(t_f)) \neq \{0\}$  alors il existe une BC-extrémale  $(\hat{x}, \hat{p}, u)$  tels que  $\hat{p}(t_f) \in N_{\mathcal{A}}(\hat{x}(t_f))$ .

**Remarque.** Cette hypothèse est assez forte, mais est vérifiée numériquement dans notre application, cf. Figure 5.11. On peut également noter que cette hypothèse est vérifiée pour les systèmes de contrôle linéaire [69, Théorème 2.1.1].

**Hypothèse H8** (Ensemble accessible).

L'ensemble accessible augmenté  $\mathcal{A}$  est un ensemble fermé non vide convexe.

#### Corollaire 4.2.10

Sous l'hypothèse (H8), si  $(\hat{x}, \hat{u})$  est une solution de (AOCP<sub>\*</sub>) alors il existe une BC-extrémale  $(\hat{x}, \hat{p}, u)$  telle que  $\hat{p}(t_f) \in N_{\mathcal{A}}(\hat{x}(t_f))$ .

► La preuve est directe. En utilisant la proposition 4.1.1, l'état final augmenté  $\hat{x}(t_f)$  appartient à  $\partial\mathcal{A}$ . D'après les propriétés 4.2.6, comme  $\mathcal{A}$  est un ensemble convexe, on a  $N_{\mathcal{A}}(\hat{x}(t_f)) \neq \{0\}$ . En utilisant le théorème 4.2.9, il existe donc une BC-extrémale  $(\hat{x}, \hat{p}, u)$  telle que  $\hat{p}(t_f) \in N_{\mathcal{A}}(\hat{x}(t_f))$ . ■

En conclusion, la démonstration du principe du maximum et les résultats précédents permettent, sous l'hypothèse (H8), de restreindre le lieu des co-états finaux au cône normal de  $\mathcal{A}$ , plus facile à déterminer que le cône de Boltyanski.

## 4.2.4 Bilan

Sous l'hypothèse (H8), il est possible de construire une BC-extrémale  $(\hat{z}, u)$  associée au problème (AOCP<sub>\*</sub>) en cherchant les solutions de

$$\begin{cases} \dot{\hat{z}}(t) = \vec{h}(t, \hat{z}(t), u(t)), & t \in [t_0, t_f] \text{ p.p.}, \\ \hat{h}(t, \hat{z}(t), u(t)) = \max_{w \in U(t)} \hat{h}(t, \hat{z}(t), w), & t \in [t_0, t_f] \text{ p.p.}, \\ \hat{z}(t_f) = \hat{z}_f, \end{cases} \quad (4.7)$$

à partir d'un couple état/co-état augmenté final  $\hat{z}_f = (\hat{x}_f, \hat{p}_f)$  tels que

$$\begin{cases} \hat{p}_f \in N_{\mathcal{A}}(\hat{x}_f), \\ \hat{p}_f \neq 0, \\ p_f^0 \leq 0, \\ x_f = x_T. \end{cases} \quad (4.8)$$

Le système (4.8) correspond aux conditions nécessaires d'optimalité du problème d'optimisation suivant

$$(O_x) \quad \begin{cases} \min_{\hat{x}_f} x_f^0, \\ \text{s.c. } \hat{x}_f \in \mathcal{A}, \\ x_f = x_T, \end{cases}$$

où  $\hat{x}_f = (x_f^0, x_f) \in \mathbb{R} \times \mathbb{R}^n$ . En effet, dans le cas général où  $\mathcal{A}$  est un ensemble fermé, on peut montrer à l'aide de [19, Théorème 10.47] que si  $\hat{x}_f$  est une solution du problème (O<sub>x</sub>), alors il existe  $\hat{p}_f \in N_{\mathcal{A}}^C(\hat{x}_f)$ , le cône normal de Clarke généralisé, tels que

$$p_f^0 \leq 0 \quad \text{et} \quad \hat{p}_f \neq 0. \quad (4.9)$$

Si de plus l'ensemble  $\mathcal{A}$  est convexe, on a  $N_{\mathcal{A}}^C(\hat{x}_f) = N_{\mathcal{A}}(\hat{x}_f)$ , ce qui mène donc directement aux conditions données par l'équation (4.8).

**Définition 4.2.11 – Point critique primal-dual**

Un point critique primal-dual du problème (O<sub>x</sub>) est un couple  $(\hat{x}_f, \hat{p}_f)$  tel que

$$\begin{aligned} \hat{x}_f &= (x_f^0, x_f) \in \mathbb{R} \times \mathbb{R}^n, & \hat{p}_f &= (p_f^0, p_f) \in \mathbb{R} \times \mathbb{R}^n, \\ \hat{x}_f &\in \mathcal{A}, & x_f &= x_T, & p_f^0 &\leq 0, & \hat{p}_f &\neq 0, & \hat{p}_f &\in N_{\mathcal{A}}(\hat{x}_f). \end{aligned}$$

Un point critique primal-dual est dit *normal* si  $p_f^0 < 0$ , et *anormal* si  $p_f^0 = 0$ .

Ces points critiques primaux-duaux du problème (O<sub>x</sub>) sont (très) fortement liées aux BC-extrémales du problème (AOCP<sub>\*</sub>). En effet, nous venons déjà de voir qu'à partir d'un point critique primal-dual du problème d'optimisation (O<sub>x</sub>), il est possible de reconstruire une BC-extrémale du problème (AOCP<sub>\*</sub>). Nous proposons donc d'introduire

l'hypothèse suivante, qui est le pendant de l'hypothèse (H4) pour le problème  $(O_x)$ , et qui nous permettra par la suite d'utiliser la première méthode de normalisation de la fonction de tir.

**Hypothèse H9** (Normalité des points critiques primaux-duaux).

Les points critiques primaux-duaux associés aux solutions de  $(O_x)$  sont normaux.

Une BC-extrémale du problème  $(OCP_*)$  est donc donnée à partir de la résolution consécutive des problèmes  $(O_x)$  et (4.7). En supposant que l'on sache résoudre le problème (4.7) (de même manière que nous l'avons fait avec l'hypothèse (H2)), les difficultés de la résolution du problème  $(OCP_*)$  sont transférées vers la connaissance de l'ensemble accessible  $\mathcal{A}$ . Le preconditionnement développé ci-dessous vise donc à transformer cet ensemble accessible en un ensemble plus facilement manipulable. En particulier, nous allons montrer que si cet ensemble accessible est la boule unité, alors on sait construire une fonction de tir affine.

### 4.3 Préconditionnement géométrique

À partir du bilan effectué dans la section 4.2.4, l'objectif est d'obtenir un point critique primal-dual du problème  $(O_x)$ , c'est-à-dire une solution du système (4.8), afin de déterminer une BC-extrémale du problème  $(AOCP_*)$ . On cherche par conséquent à proposer une méthode qui permettra une résolution efficace du problème  $(O_x)$ . Cette méthode correspond à une méthode de tir simple utilisant le preconditionnement géométrique développée ci-dessous.

#### 4.3.1 Principe

Le principe de ce preconditionnement est de construire un changement de variables d'état augmenté qui permet de transformer le problème  $(O_x)$  en un problème  $(O_y)$  de la même forme et plus facile à résoudre. Une fois le problème résolu dans les variables transformées, on obtient les états et co-états augmentés solutions dans les variables d'origine par un relèvement adéquat.

##### 4.3.1.1 Caractéristiques du changement de variables

Pour tout difféomorphisme  $\phi$  de  $\mathbb{R}^{n+1}$ , le problème d'optimisation  $(O_x)$  se réécrit dans les nouvelles coordonnées par

$$(\tilde{O}_y) \quad \begin{cases} \min_{\hat{y}_f} (\phi^{-1})_0(\hat{y}_f) \\ \text{s.c. } \hat{y}_f \in \phi(\mathcal{A}), \\ (\phi^{-1})_+(\hat{y}_f) = x_T. \end{cases}$$

où

$$\phi(\hat{x}_f) = (\phi_0(\hat{x}_f), \phi_+(\hat{x}_f)) \quad \text{et} \quad \phi^{-1}(\hat{y}_f) = ((\phi^{-1})_0(\hat{y}_f), (\phi^{-1})_+(\hat{y}_f)),$$

dans le sens où  $\hat{x}_f$  est une solution de  $(O_x)$  si et seulement si  $\hat{y}_f = \phi(\hat{x}_f)$  est une solution de  $(\tilde{O}_y)$ . Les fonctions  $\phi_0$  et  $(\phi^{-1})_0$  sont définies de  $\mathbb{R}^{n+1}$  à valeurs dans  $\mathbb{R}$  et ne sont pas

des difféomorphismes. La notation utilisée pour  $(\phi^{-1})_0$  permet de souligner que cette fonction n'est pas l'inverse de  $\phi_0$ . Cette remarque et ces notations sont aussi valables pour les fonctions  $\phi_+$  et  $(\phi^{-1})_+$ .

Nous allons montrer que la condition

$$\frac{\partial \phi}{\partial x^0} = k e_1, \quad \text{avec } k > 0 \quad \text{et} \quad e_1 = (1, 0, \dots, 0) \in \mathbb{R}^{n+1}, \quad (4.10)$$

sur le difféomorphisme  $\phi$  permet de conférer au problème  $(\tilde{O}_y)$  la même forme que le problème  $(O_x)$  et de le rendre équivalent au problème suivant :

$$(O_y) \quad \begin{cases} \min_{\hat{y}_f} y_f^0 \\ \text{s.c. } \hat{y}_f \in \phi(\mathcal{A}), \\ y_f = y_T, \end{cases}$$

où  $y_T$  est défini juste après, équation (4.11). En effet, si  $\phi$  satisfait (4.10), alors la fonction  $\phi_+$  ne dépend pas de  $x^0$ , tout comme la fonction  $(\phi^{-1})_+$  ne dépend pas de  $y^0$ . On propose de nommer

$$\varphi_+ : \mathbb{R}^n \rightarrow \mathbb{R}^n \quad \text{et} \quad (\varphi^{-1})_+ : \mathbb{R}^n \rightarrow \mathbb{R}^n$$

les restrictions aux variables dépendantes  $x$  (respectivement  $y$ ) des fonctions  $\phi_+$  (resp.  $(\phi^{-1})_+$ ). On peut montrer que ces deux fonctions sont des difféomorphismes, et sont inverses l'une de l'autre. C'est pourquoi on note maintenant  $\varphi_+^{-1}$  à la place de  $(\varphi^{-1})_+$ . L'hyperplan  $x_f = x_T$  est transformé dans les nouvelles coordonnées en  $y_f = y_T$ , où

$$y_T = \varphi_+(x_T). \quad (4.11)$$

Définissons maintenant les fonctions  $\varphi_0 : \mathbb{R} \rightarrow \mathbb{R}$  et  $(\varphi^{-1})_0 : \mathbb{R} \rightarrow \mathbb{R}$  par

$$\varphi_0(x_f^0) = \phi_0(x_f^0, x_T) \quad \text{et} \quad (\varphi^{-1})_0(y_f^0) = (\phi^{-1})_0(y_f^0, y_T),$$

comme les restrictions des fonctions  $\phi_0$  et  $(\phi^{-1})_0$  aux droites  $x = x_T$  et  $y = y_T$ . À nouveau, on peut montrer que ces deux fonctions sont des difféomorphismes de  $\mathbb{R}$  et sont inverses l'une de l'autre. On propose donc comme précédemment de noter  $\varphi_0^{-1}$  au lieu de  $(\varphi^{-1})_0$ . On remarque alors que l'équation (4.10) nous donne

$$\varphi_0' > 0 \quad \text{et} \quad (\varphi_0^{-1})' > 0,$$

ce qui implique que

$$(\tilde{O}_y) \iff \begin{cases} \min_{\hat{y}_f} \varphi_0^{-1}(y_f^0), \\ \text{s.c. } \hat{y}_f \in \phi(\mathcal{A}), \\ y_f = \varphi_+(x_T), \end{cases} \iff \begin{cases} \min_{\hat{y}_f} y_f^0, \\ \text{s.c. } \hat{y}_f \in \phi(\mathcal{A}), \\ y_f = y_T, \end{cases}$$

qui est bien le problème  $(O_y)$  attendu.



**Remarque 4.3.1.** De la même manière que nous venons de montrer que  $(O_x)$  est équivalent à  $(O_y)$ , il est possible de montrer sous les mêmes hypothèses que le problème de commande optimale  $(AOCP_*)$  est équivalent au problème de commande optimale suivant

$$\begin{cases} \min_{\hat{y}, u} y^0(t_f) \\ \text{s.c. } \dot{\hat{y}}(t) = \hat{f}_{\hat{y}}(t, \hat{y}(t), u(t)), & t \in [t_0, t_f] \text{ p.p.}, \\ u(t) \in U(t), & t \in [t_0, t_f], \\ \hat{y}(t_0) = \phi(\hat{x}_0) \quad y(t_f) = y_T, \end{cases}$$

où  $f_{\hat{y}}(t, \hat{y}, u) = J_{\phi^{-1}}(\hat{y})^{-1} \hat{f}(t, \phi^{-1}(\hat{y}), u)$ .

#### 4.3.1.2 Lien entre les variables duales - Transformée de Mathieu

Maintenant que nous avons construit une correspondance  $\phi$  entre les états augmentés  $\hat{x}$  et  $\hat{y}$ , il faut établir un lien similaire entre les variables duales. Pour ce faire, intéressons-nous aux conditions nécessaires d'optimalité du problème  $(O_y)$  et établissons le lien avec celles du problème  $(O_x)$  données par l'équation (4.8). De la même manière que nous l'avons fait pour le problème  $(O_x)$ , en utilisant [19, Théorème 10.47], si  $\hat{y}$  est solution de  $(O_y)$  alors il existe  $\hat{q}_f = (q_f^0, q_f) \in N_{\phi(\mathcal{A})}^C(\hat{y}_f)$  tel que

$$q_f^0 \leq 0 \quad \text{et} \quad \hat{q}_f \neq 0.$$

**Remarque.** Dans ce qui suit, nous aurons toujours  $\phi(\mathcal{A})$  convexe,

- soit parce que  $\phi$  est affine et  $\mathcal{A}$  convexe ;
- soit parce que  $\phi(\mathcal{A})$  est la boule unité donc convexe.

Il en résulte que  $\hat{q}_f = (q_f^0, q_f) \in N_{\phi(\mathcal{A})}(\hat{y}_f)$ .

Le lien entre la variable duale  $\hat{q}_f \in N_{\phi(\mathcal{A})}(\hat{y}_f)$  associée au problème  $(O_x)$  et la variable duale  $\hat{p}_f \in N_{\mathcal{A}}(\hat{x}_f)$  associée au problème  $(O_y)$  est donnée directement par le corollaire 4.2.8 :

$$\hat{q}_f = J_{\phi}(\hat{x}_f)^{-\top} \hat{p}_f \quad \text{et} \quad \hat{p}_f = J_{\phi^{-1}}(\hat{y}_f)^{-\top} \hat{q}_f.$$

Finalement, le relèvement de  $\phi$  dans l'espace primal-dual est donné par

$$\Phi(\hat{x}_f, \hat{p}_f) = \left( \phi(\hat{x}_f), J_{\phi}(\hat{x}_f)^{-\top} \hat{p}_f \right). \quad (4.12)$$

**Remarque 4.3.2.** D'un point de vue commande optimale, c'est-à-dire si on considère le problème  $(AOCP_*)$  et son pendant dans les nouvelles coordonnées, le relèvement dans l'espace des phases du difféomorphisme  $\phi$  donné par  $\Phi$  est appelé la transformée de Mathieu. En effet, la transformée de Mathieu est un symplectomorphisme, c'est-à-dire un difféomorphisme sur l'espace co-tangent qui préserve la 2-forme canonique des variétés symplectiques dont la dynamique hamiltonienne est issue. Une courte description des variétés symplectiques, de l'origine hamiltonienne ainsi qu'une preuve que la transformée de Mathieu est un symplectomorphisme est donnée dans l'annexe A.

### 4.3.1.3 Mise en œuvre progressive

Nous venons de décrire dans la section 4.3.1.1 les caractéristiques du changement de variable  $\phi$  afin que le problème dans les nouvelles coordonnées ait la même formulation que le problème  $(O_x)$ . De plus, nous avons donné dans la section 4.3.1.2 la relation entre les variables duales, c'est-à-dire le relèvement canonique donnée par la transformée de Mathieu dans l'espace primal-dual à partir d'une transformation sur l'espace primal. L'objectif est maintenant de montrer comment un choix judicieux du changement de variable  $\phi$  permet de rendre la fonction de tir mieux adaptée à l'utilisation d'un solveur de type Newton pour la résolution du problème transformé. Pour cela, nous allons considérer les trois cas suivants, associés à des hypothèses de moins en moins restrictives :

- cas 1 : il existe un difféomorphisme entre  $\mathcal{A}$  et la boule unité,
- cas 2 : l'ensemble  $\mathcal{A}$  est connu sous forme fermée,
- cas général : l'ensemble accessible est donné par l'équation (4.2).

La méthode de résolution proposée dans chacun de ces cas s'appuie sur la résolution du précédent, sauf pour le Cas 1 dont la résolution est assez simple. De plus, chacun de ces cas est illustré par un exemple.

### 4.3.2 Cas 1 : l'ensemble accessible est difféomorphe à la boule

Dans ce premier cas, on fait l'hypothèse que l'on sait construire un difféomorphisme  $\phi$  qui vérifie la condition (4.10) entre l'ensemble accessible augmenté  $\mathcal{A}$  et  $\mathcal{B}^{n+1}$ . L'objectif principal de cette section est de montrer que dans ce cas on sait facilement résoudre le problème  $(O_y)$  (et donc le problème  $(O_x)$ ). De plus, la fonction de tir  $T_2$  dans les nouvelles variables est une fonction affine.

**Résolution.** Comme  $\phi(\mathcal{A}) = \mathcal{B}^{n+1}$ , le problème  $(O_y)$  s'écrit alors

$$(O_{y_{\mathcal{B}}}) \quad \begin{cases} \min_{\hat{y}_f} y_f^0, \\ \text{s.c. } \hat{y}_f \in \mathcal{B}^{n+1}, \\ y_f = y_T. \end{cases}$$

Sa solution est trivialement donnée par  $\hat{y}_f = (\eta(y_T), y_T)$ , où la fonction  $\eta: \mathcal{B}^n \rightarrow \mathbb{R}^-$  est définie par

$$\eta(e) = -\sqrt{1 - \|e\|_2^2}.$$

On cherche maintenant à exprimer les points de la frontière de l'ensemble accessible en fonction du vecteur normal associé. Ici, la frontière de l'ensemble accessible correspond au cercle unité

$$\mathcal{C}^{n+1} = \left\{ \hat{y}_f \in \mathbb{R}^{n+1} \mid \|\hat{y}_f\|_2 = 1 \right\}.$$

Le cône normal de  $\mathcal{B}^{n+1}$  au point  $\hat{y}_f \in \mathcal{C}^{n+1}$  est donné par

$$N_{\mathcal{B}^{n+1}}(\hat{y}_f) = \{ \lambda \hat{y}_f \mid \lambda \geq 0 \}.$$

On en déduit que tout vecteur non nul  $\hat{q}_f$  de  $\mathbb{R}^{n+1}$  est normal extérieur à  $\mathcal{B}^{n+1}$  en un point  $\hat{y}_f \in \mathcal{C}^{n+1}$  défini par

$$\hat{y}_f = \hat{y}_f(\hat{q}_f) = \frac{\hat{q}_f}{\|\hat{q}_f\|_2}.$$

En notant  $\hat{y}_f(\cdot) = (y_f^0(\cdot), y_f(\cdot))$ , on a donc

$$y_f^0(\hat{q}_f) = \frac{q_f^0}{\|\hat{q}_f\|_2} \quad \text{et} \quad y_f(\hat{q}_f) = \frac{q_f}{\|\hat{q}_f\|_2}.$$

Le système (4.8) appliqué au problème  $(\mathbf{O}_{y_B})$  peut être résolu par une méthode de tir en cherchant un zéro non trivial de la fonction  $T: \mathbb{R}^- \times \mathbb{R}^n \rightarrow \mathbb{R}^n$  définie par

$$T(\hat{q}_f) = y_f(\hat{q}_f) - y_T.$$

En effet, si  $\hat{q}_f \neq 0$  est solution de  $T(\cdot) = 0$ , alors le couple  $(\hat{y}_f(\hat{q}_f), \hat{q}_f)$  est un point critique primal-dual de  $(\mathbf{O}_{y_B})$ . Dans ce cas trivial, le point  $\hat{y}_f$  est également une solution de  $(\mathbf{O}_{y_B})$ . De plus, pour tout  $k > 0$ , le point  $k\hat{q}_f$  est également un zéro non trivial de la fonction  $T$ . De la même manière qu'au chapitre 2, pour casser cette homogénéité, on propose deux méthodes de normalisation qui mènent à deux fonctions de tir différentes. La première consiste à se placer sous l'hypothèse (H9) qui nous permet de fixer  $q_f^0 = -1$ . La fonction de tir associée  $T_1: \mathbb{R}^n \rightarrow \mathbb{R}^n$  est alors définie par

$$T_1(q_f) = T(-1, q_f) = \frac{q_f}{\|(-1, q_f)\|_2} - y_T.$$

La seconde méthode proposée n'a pas besoin de cette hypothèse. L'idée est de fixer  $\|\hat{q}_f\| = 1$ , ce qui mène à la fonction de tir  $T_2: \mathcal{B}^n \rightarrow \mathbb{R}$  définie par

$$T_2(q_f) = T(\eta(q_f), q_f).$$

On a donc

$$T_2(q_f) = \frac{q_f}{\|(\eta(q_f), q_f)\|_2} - y_T = q_f - y_T.$$

Finalement, si l'ensemble accessible du système augmenté est  $\mathcal{B}^{n+1}$ , la fonction de tir  $T_2$  est affine, ce qui permettrait à notre solveur de type Newton de converger en une seule itération à partir d'une initialisation dans la boule  $\mathcal{B}^n$ . À vrai dire, dans ce cas trivial, la racine de  $T_2$  est même directement fournie par  $q_f = y_T$ .

En utilisant la transformée de Mathieu donnée par l'équation (4.12) introduite dans la section 4.3.1.2, un point critique primal-dual  $(\hat{x}_f, \hat{p}_f)$  associée à la solution de  $(\mathbf{O}_x)$  est donnée à partir du zéro  $q_f = y_T$  de la fonction de tir par

$$(\hat{x}_f, \hat{p}_f) = \Phi^{-1}(\hat{y}_f, \hat{q}_f) = (\phi^{-1}(\hat{y}_f), J_{\phi^{-1}}(\hat{y}_f)^{-\top} \hat{q}_f), \quad (4.13)$$

où  $\hat{q}_f = (\eta(q_f), q_f)$  et  $\hat{y}_f = \hat{q}_f$ .

**Exemple.** On considère ici un cas particulier du cas 1 où la frontière de l'ensemble accessible  $\mathcal{A} \subset \mathbb{R}^2$  est une ellipse. L'objectif est donc de construire le difféomorphisme  $\phi$  qui satisfait l'équation (4.10) afin d'appliquer la méthode de résolution proposée précédemment. Le problème d'optimisation considéré correspond donc au problème

$$(O_{x_{\mathcal{E}}}) \quad \begin{cases} \min_{\hat{x}_f} x_f^0 \\ \text{s.c. } \hat{x}_f \in \mathcal{E}, \\ x_f = x_T. \end{cases}$$

L'ellipse  $\partial\mathcal{E}$  est géométriquement définie par son centre  $c = (c_0, c_1) \in \mathbb{R}^2$ , ses demi-axes  $a, b > 0$  et l'angle  $\theta \in [0, \pi/2[$  entre l'axe portant la deuxième composante de  $\hat{x}$  (horizontal sur la figure 4.4) et le demi-axe de longueur  $b$ . On a donc

$$\mathcal{E} = r(-\theta)s(a, b)\mathcal{B} + c$$

où  $\mathcal{B} \subset \mathbb{R}^2$  correspond à la boule unité fermée pour la norme euclidienne de dimension 2, et où

$$r(\theta) = \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix} \quad \text{et} \quad s(a, b) = \begin{bmatrix} a & 0 \\ 0 & b \end{bmatrix}$$

correspondent respectivement à la matrice de rotation d'angle  $\theta$  de  $\mathbb{R}^2$  et à la matrice de mise à l'échelle. La géométrie de l'ellipse et les paramètres correspondants sont représentés sur la figure 4.4.

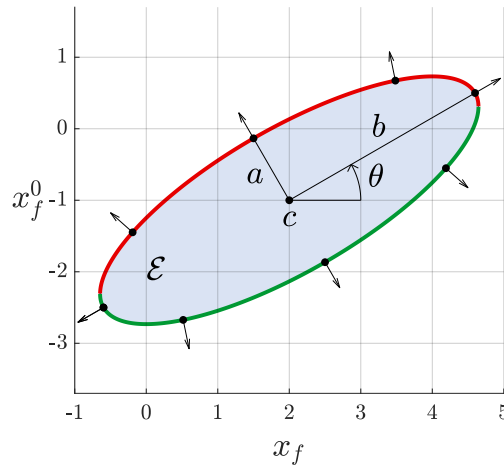


FIGURE 4.4 – Ellipse  $\mathcal{E}$  avec  $a = 1$ ,  $b = 3$ ,  $c = (-1, 2)$  et  $\theta = \pi/6$ .

Une équation paramétrique de  $\partial\mathcal{E}$  est donc donnée pour tout  $\beta \in [-\pi, \pi[$  par

$$\hat{x}_f(\beta) = r(-\theta)s(a, b) \begin{bmatrix} \sin \beta \\ \cos \beta \end{bmatrix} + c.$$

Cette écriture associe tacitement le point  $(0, 1)$  du cercle au point  $(b \sin \theta, b \cos \theta)$  de l'ellipse, mais on peut la généraliser pour y associer un autre point par l'introduction

d'un paramètre de déphasage  $\beta_0 \in [-\pi, \pi[$  :

$$\hat{x}_f(\beta) = r(-\theta)s(a, b) \begin{bmatrix} \sin(\beta - \beta_0) \\ \cos(\beta - \beta_0) \end{bmatrix} + c,$$

qu'on peut aussi écrire

$$\begin{bmatrix} \sin \beta \\ \cos \beta \end{bmatrix} = \phi_{\beta_0}(\hat{x}_f(\beta)) \quad \text{avec} \quad \phi_{\beta_0}(\hat{x}_f) = r(-\beta_0)s(a^{-1}, b^{-1})r(\theta)(\hat{x}_f - c). \quad (4.14)$$

Parmi les difféomorphismes  $\phi_{\beta_0}$  de la forme (4.14) qui permettent de transformer l'ellipse en cercle unité, on s'attache maintenant à construire celui qui permet de transformer le problème  $(O_x)$  en le problème  $(O_{y_B})$ , c'est-à-dire celui qui satisfait de l'équation (4.10). Pour  $\beta_0$  fixé, le difféomorphisme  $\phi_{\beta_0}$  est affine, et sa jacobienne est donnée par  $r(-\beta_0)s(a^{-1}, b^{-1})r(\theta)$  dont la première colonne s'écrit

$$\frac{\partial \phi_{\beta_0}}{\partial x^0} = \begin{bmatrix} \frac{\cos \beta_0 \cos \theta}{a} + \frac{\sin \beta_0 \sin \theta}{b} \\ -\frac{\sin \beta_0 \cos \theta}{a} + \frac{\cos \beta_0 \sin \theta}{b} \end{bmatrix}.$$

Puisque  $a, b > 0$ , l'équation (4.10) s'écrit

$$\begin{cases} b \cos \beta_0 \cos \theta + a \sin \beta_0 \sin \theta > 0, \\ -b \sin \beta_0 \cos \theta + a \cos \beta_0 \sin \theta = 0. \end{cases}$$

Comme  $\theta \in [0, \pi/2[$ ,  $\cos \theta > 0$  et en posant  $d = \frac{a \sin \theta}{b \cos \theta} > 0$ , le système précédent s'écrit encore

$$\begin{cases} \cos \beta_0 + d \sin \beta_0 > 0, \\ -\sin \beta_0 + d \cos \beta_0 = 0. \end{cases}$$

On remarque que  $\beta_0 = \pm\pi/2$  ne peut pas être solution de la deuxième équation, donc  $\cos \beta_0$  est nécessairement non nul et on peut réécrire le système de la manière suivante :

$$\begin{cases} (1 + d^2) \cos \beta_0 > 0, \\ \sin \beta_0 = d \cos \beta_0, \end{cases}$$

ou encore

$$\begin{cases} \cos \beta_0 > 0, \\ \tan \beta_0 = d. \end{cases}$$

dont l'unique solution dans  $\in [-\pi, \pi[$  est  $\beta_0^* = \arctan(d)$ .

Finalement, nous venons de construire le difféomorphisme linéaire  $\phi = \phi_{\beta_0^*}$  qui transforme l'ellipse  $\partial\mathcal{E}$  en cercle  $\partial\mathcal{B}$ , tout en respectant l'équation (4.10). Comme nous l'avons établi au paragraphe précédent, le problème  $(O_{x_{\mathcal{E}}})$  est équivalent par  $\phi$  au problème  $(O_{y_{\mathcal{B}}})$  (avec  $n = 1$ ). L'image de l'ensemble  $\mathcal{E}$  par le difféomorphisme  $\phi$  est donnée dans la figure 4.5b.

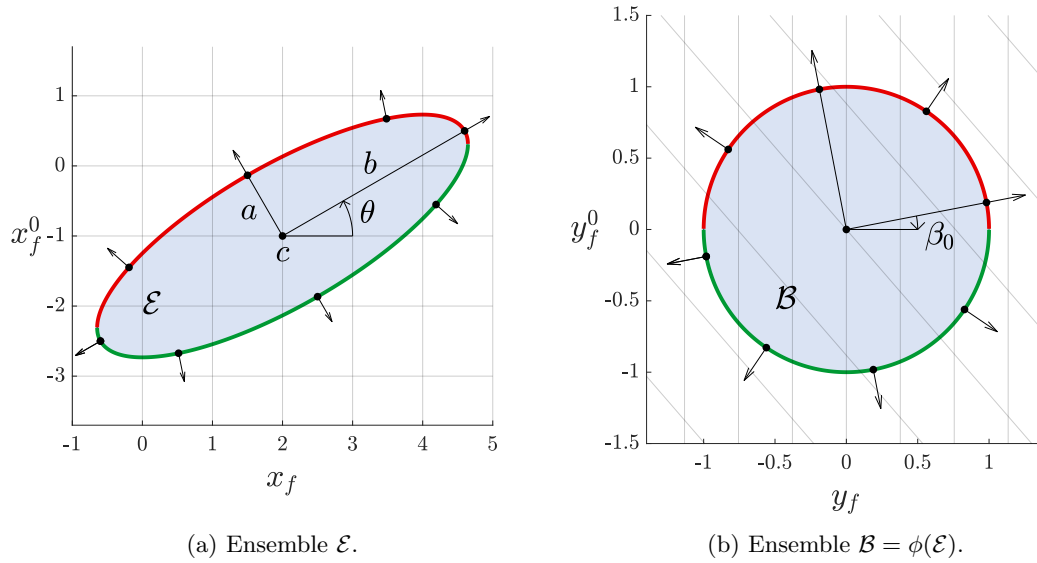


FIGURE 4.5 – Ensembles  $\mathcal{E}$  et  $\mathcal{B} = \phi(\mathcal{E})$ , avec  $a = 1$ ,  $b = 3$ ,  $c = (-1, 2)$  et  $\theta = \pi/6$ . Le repère orthonormé (en gris) de la figure de gauche a également été transformé par  $\phi$  sur la figure de droite, afin de mettre l'accent sur le fait que  $\phi$  transforme la droite  $x = x_T$  en  $y = y_T$  par construction

### 4.3.3 Cas 2 : extension au cas non difféomorphe

Dans la section précédente, nous avons présenté une méthode de résolution efficace du problème  $(O_x)$  s'il est possible d'exhiber un difféomorphisme entre l'ensemble accessible du système augmenté  $\mathcal{A}$  et la boule unité  $\mathcal{B}^{n+1}$  qui satisfait (4.10). Cette hypothèse est encore trop restrictive. En effet, il peut être difficile - voire impossible - de trouver un tel difféomorphisme dans certaines applications.

Dans cette section, on cherche toujours une solution primale-duale du problème

$$(O_x) \quad \begin{cases} \min_{\hat{x}_f} x_f^0 \\ \text{s.c. } \hat{x}_f \in \mathcal{A}, \\ x_f = x_T, \end{cases}$$

mais cette fois en supposant que

- l'on sait construire des points de la frontière de l'ensemble accessible  $\mathcal{A}$ ,
- et que l'on connaît une fonction qui à tout vecteur  $\hat{p}_f \in \mathbb{R}^{n+1}$  non trivial associe un point  $\hat{x}_f \in \partial\mathcal{A}$  tel que  $\hat{p}_f \in N_{\mathcal{A}}(\hat{x}_f)$ . On note cette fonction

$$\hat{x}_f : \mathbb{R}^{n+1} \longrightarrow \partial\mathcal{A} \subset \mathbb{R} \times \mathbb{R}^n \\ \hat{p}_f \longmapsto \hat{x}_f(\hat{p}_f) = (x_f^0(\hat{p}_f), x_f(\hat{p}_f)). \quad (4.15)$$

### 4.3.3.1 Méthode de résolution classique

La recherche d'un point critique primal-dual du problème  $(O_x)$  peut être effectuée en cherchant un zéro non trivial de la fonction de tir  $S: \mathbb{R}^- \times \mathbb{R}^n \rightarrow \mathbb{R}^n$  définie par

$$S(\hat{p}_f) = x_f(\hat{p}_f) - x_T.$$

De plus, nous pouvons utiliser les deux méthodes de normalisation proposées précédemment. En effet, sous l'hypothèse (H9), on peut fixer  $p_f^0 = -1$  et chercher un zéro de  $S_1: \mathbb{R}^n \rightarrow \mathbb{R}^n$  définie par

$$S_1(p_f) = S(-1, p_f).$$

Sans cette hypothèse, on peut aussi fixer  $\|\hat{p}_f\|_2 = 1$  et s'intéresser à la fonction de tir  $S_2: \mathcal{B}^n \rightarrow \mathbb{R}^n$  définie par

$$S_2(p_f) = S(\eta(p_f), p_f).$$

Numériquement, on utilise un solveur de type Newton pour trouver un zéro de  $S_1$  ou de  $S_2$ . Cette recherche peut être assez complexe, surtout si ces fonctions de tirs sont mal conditionnées, notamment

- si l'ensemble accessible a de mauvaises caractéristiques, par exemple une différence d'échelle importante entre le coût et l'état qui peut mener à une plage d'évolution de la fonction de tir relativement restreinte, rendant de ce fait l'initialisation du solveur assez complexe ;
- et/ou si la fonction de tir est fortement non linéaire, ce qui risque d'induire un nombre important d'itérations du solveur.

Dans ce qui suit, nous proposons une méthode, principalement basée sur les idées développées dans la section 4.3.2, qui permet de construire des fonctions de tir qui ont de fortes chances d'être mieux conditionnées.

### 4.3.3.2 Méthode de tir avec preconditionnement

Nous avons vu avec le cas 1 que la résolution du problème  $(O_x)$  est particulièrement simple lorsque l'on connaît un changement de variables  $\phi$  correctement choisi qui transforme  $\mathcal{A}$  en la boule unité, ce qui est le cas dans l'exemple considéré lorsque la frontière de  $\mathcal{A} \subset \mathbb{R}^2$  est une ellipse  $\mathcal{E}$ . Ici, on ne sait a priori pas construire un tel changement de variables mais on sait construire la frontière de  $\mathcal{A} \subset \mathbb{R}^{n+1}$ . On va donc approcher celle-ci par une  $(n+1)$ -ellipse et se ramener au cas précédent. La fonction de tir que l'on obtiendra n'aura alors plus de raison d'être affine mais on peut espérer qu'elle s'en rapproche si l'approximation de  $\mathcal{A}$  par  $\mathcal{E}$  est suffisamment bonne.

**Remarque.** Le choix d'approcher  $\mathcal{A}$  par une  $(n+1)$ -ellipse est justifié par l'allure de l'ensemble accessible de l'application qui a motivé ce travail (avec  $n = 1$ ) et qui est représenté dans la figure 5.11a. Des approches basées sur l'approximation de l'ensemble accessible par des ellipsoïdes (contenue par  $\mathcal{A}$  ou contenant  $\mathcal{A}$ ) ont été développés dans la littérature [42, 57] et ont pour objectif d'obtenir des informations sur les systèmes étudiés. Il est par exemple possible d'avoir une estimation des bornes minimum et maximum du coût optimal.

**Approximation de l'ensemble accessible par une ellipse.** Comme nous supposons avoir accès à des points de la frontière de l'ensemble  $\mathcal{A}$ , il est possible de l'approcher par une  $(n+1)$ -ellipse  $\mathcal{E}^{n+1}$ , qui correspond à la généralisation de l'ellipse en dimension  $n+1$ . Une telle ellipse peut être décrite de la façon suivante

$$\mathcal{E}^{n+1} = (r^{n+1}(\theta))^{-1} s^{n+1}(a) \mathcal{B}^{n+1} + c$$

à partir :

- de son centre  $c \in \mathbb{R}^{n+1}$ ,
- de ses demi-axes  $a \in (\mathbb{R}^+)^{n+1}$  et de la matrice de mise à l'échelle associée qui n'est rien d'autre que la matrice diagonale générée par le vecteur  $a$ ,
- et de son orientation caractérisée par  $\frac{n(n+1)}{2}$  paramètres  $\theta$  et la matrice de rotation associée  $r^{n+1}(\theta)$ .

L'approximation de  $\mathcal{A}$  par  $\mathcal{E}^{n+1}$  est alors obtenue par ajustement des paramètres  $c$ ,  $a$  et  $\theta$ , par exemple au moyen d'une méthode de moindres carrés.

**Construction du difféomorphisme.** De la même manière que nous l'avons fait dans le cas  $n=1$  dans la section 4.3.2, il est possible de trouver un difféomorphisme  $\phi$  qui satisfait l'équation (4.10) dans la famille des difféomorphismes linéaires générés par

$$\phi_\gamma(\hat{x}_f) = r^{n+1}(\gamma) (s^{n+1}(a))^{-1} r^{n+1}(\theta) (\hat{x}_f - c),$$

qui transforment  $\mathcal{E}^{n+1}$  en  $\mathcal{B}^{n+1}$  par construction.

**Remarque.** L'expression d'une matrice de rotation en dimension  $n+1$  ainsi que la méthode pour obtenir le difféomorphisme  $\phi$  en dimension  $n+1$  ne seront pas détaillées. En effet, nous nous intéressons dans cette thèse à la résolution de problèmes où  $n=1$ . Cependant, nous voulons mettre en avant que la méthode proposée est généralisable en dimension  $n+1$  sans introduire de difficulté théorique supplémentaire.

Puisque  $\phi$  est affine, on note

$$\phi(\hat{x}) = A\hat{x} + b,$$

où  $A$  est une matrice carrée de dimension  $n+1$  et  $b$  un vecteur de  $\mathbb{R}^{n+1}$ . De plus, comme le difféomorphisme  $\phi$  satisfait l'équation (4.10), la matrice  $A$  et le vecteur  $b$  s'expriment sous la forme suivante

$$A = \begin{bmatrix} k & A_{x^0} \\ 0 & A_x \end{bmatrix} \quad \text{et} \quad b = \begin{bmatrix} b_{x^0} \\ b_x \end{bmatrix}$$

où  $k > 0$ ,  $A_{x^0}$  est un vecteur ligne de dimension  $n$ ,  $A_x$  est une matrice carrée de dimension  $n$ ,  $b_{x^0} \in \mathbb{R}$  et  $b_x$  est un vecteur colonne de dimension  $n$ . Comme nous l'avons vu dans la section 4.3.1.1, le problème  $(O_x)$  est équivalent par  $\phi$  au problème

$$(O_y) \quad \begin{cases} \min_{\hat{y}_f} y_f^0 \\ \text{s.c. } \hat{y}_f \in \phi(\mathcal{A}), \\ y_f = y_T, \end{cases}$$



où  $y_T = A_x x_T + b_x$ . Cependant, par rapport à la section 4.3.2, l'ensemble  $\phi(\mathcal{A})$  n'est plus la boule unité, mais un ensemble que l'on espère assez proche puisque  $\mathcal{E}^{n+1}$  n'est qu'une approximation de  $\mathcal{A}$ . On remarque cependant que l'ensemble  $\phi(\mathcal{A})$  est un ensemble fermé convexe, puisque  $\mathcal{A}$  est un ensemble fermé convexe par hypothèse et que  $\phi$  est un difféomorphisme linéaire.

**Construction de la nouvelle fonction de tir.** Nous allons maintenant construire le pendant de la fonction  $\hat{x}_f(\cdot)$  dans les nouvelles coordonnées, c'est-à-dire la fonction  $\hat{y}_f: \mathbb{R}^{n+1} \rightarrow \mathbb{R}^{n+1}$  qui à un vecteur  $\hat{q}_f \in \mathbb{R}^{n+1}$  non trivial associe  $\hat{y}_f \in \partial(\phi(\mathcal{A}))$  tel que  $\hat{q}_f \in N_{\phi(\mathcal{A})}(\hat{y}_f)$ . Cette fonction peut être construite à l'aide de la fonction  $\hat{x}_f(\cdot)$ , que l'on connaît par la seconde hypothèse faite au début de cette section 4.3.3, de la transformée de Mathieu inverse (4.13) et du difféomorphisme  $\phi$  :

$$\begin{aligned}\hat{y}_f(\hat{q}_f) &= \phi\left(\hat{x}_f(J_{\phi^{-1}}(\hat{x}_f)^{-\top} \hat{q}_f)\right) \\ &= A\hat{x}_f(A^\top \hat{q}_f) + b.\end{aligned}$$

En notant  $M(\cdot)$  le relèvement en co-état/variable duale donnée par la transformée de Mathieu (4.12), qui s'exprime par  $M(\hat{p}_f) = J_\phi(\hat{x}_f)^{-\top} \hat{p}_f = A^{-\top} \hat{p}_f$  dans notre cas affine, une représentation schématiquement la fonction  $\hat{y}_f(\cdot)$  est donnée par la figure 4.6.

$$\begin{array}{ccc} \hat{p}_f & \xleftarrow[\hat{p}_f = A^\top \hat{q}_f]{\hat{p}_f = M^{-1}(\hat{q}_f)} & \hat{q}_f \\ \hat{x}_f(\hat{p}_f) \downarrow & & \\ \hat{x}_f & \xrightarrow[A\hat{x}_f + b = \hat{y}_f]{\phi(\hat{x}_f) = \hat{y}_f} & \hat{y}_f \end{array}$$

FIGURE 4.6 – Schéma de la construction de la fonction  $\hat{y}_f(\cdot)$  dans le Cas 2.

En notant  $\hat{y}_f(\hat{q}_f) = (y_f^0(\hat{q}_f), y_f(\hat{q}_f))$ , on a directement une expression de  $y_f: \mathbb{R}^{n+1} \rightarrow \mathbb{R}^n$  donnée par

$$y_f(\hat{q}_f) = A_x x_f(A^\top \hat{q}_f) + b_x.$$

La fonction de tir dans les nouvelles coordonnées  $T: \mathbb{R}^{n+1} \rightarrow \mathbb{R}^n$  est donc définie par

$$T(\hat{q}_f) = y_f(\hat{q}_f) - y_T,$$

ce qui, sous l'hypothèse (H9), nous permet de construire la fonction de tir  $T_1: \mathbb{R}^n \rightarrow \mathbb{R}^n$  définie par

$$T_1(q_f) = T(-1, q_f), \quad (4.16)$$

ainsi que, sans cette hypothèse, la fonction de tir  $T_2: \mathcal{B}^n \rightarrow \mathbb{R}^n$  définie par

$$T_2(q_f) = T(\eta(q_f), q_f). \quad (4.17)$$

On rappelle que si  $\mathcal{A}$  était l'ellipse  $\mathcal{E}^{n+1}$ , la fonction de tir  $T_2$  serait la fonction affine  $q_f \mapsto q_f - y_T$  définie sur  $\mathcal{B}^n$ , et un zéro de  $T_2$  serait donc donné par  $q_f = y_T$ . De

plus, un solveur de type Newton trouverait cette solution en une seule itération à partir d'une initialisation sur  $\mathcal{B}^n$ . Cependant dans notre cas, l'ensemble accessible du système augmenté  $\mathcal{A}$  n'est pas l'ellipse  $\mathcal{E}^{n+1}$ , mais un ensemble qu'on espère assez proche. On s'attend donc à réduire le nombre d'itérations en utilisant la fonction  $T_2$  plutôt que les fonctions de tir dans les coordonnées initiales. De plus, le point  $q_f = y_T$  peut constituer une bonne initialisation pour le solveur.

### 4.3.3.3 Exemple

Pour illustrer la méthode proposée, nous allons l'appliquer au problème (E) introduit dans la section 2.1.3. Il est possible de montrer en s'appuyant sur les résultats de l'exemple présenté en page 29 que l'ensemble accessible augmenté s'exprime de la manière suivante

$$\mathcal{A} = \left\{ (x_f^0, x_f) \left| \begin{array}{l} x_f \in [-t_f, t_f], \\ x_f^0 \geq \frac{1}{4}(x_f^2 + 2x_f t_f - t_f^2), \\ x_f^0 \leq \frac{1}{4}(t_f^2 + 2x_f t_f - x_f^2) \end{array} \right. \right\}.$$

Cet ensemble est représenté sur la figure 4.7a. Comme nous pouvons le constater, cet ensemble est convexe fermé et donc tout vecteur non trivial  $\hat{p}_f \in \mathbb{R}^2$  peut être rattaché de manière normale à un point  $\hat{x}_f \in \partial\mathcal{A}$ . La fonction  $\hat{x}_f(\cdot)$  décrite par l'équation (4.15) est définie pour tout  $\hat{p}_f$  non trivial par

$$\hat{x}_f(\hat{p}_f) = (x_f^0(\hat{p}_f), x_f(\hat{p}_f)),$$

où, en posant  $t_b = \frac{p_f}{p^0}$  (pour tout  $p^0 \neq 0$ ), les fonctions  $x_f(\cdot)$  et  $x_f^0(\cdot)$  sont définies respectivement par

$$x_f(p^0, p_f) = \begin{cases} t_f \operatorname{signe}(p_f) & \text{si } p^0 = 0, \\ t_f \operatorname{signe}(p^0) & \text{si } p^0 \neq 0 \text{ et } t_b > 0, \\ (2t_b + t_f) \operatorname{signe}(p^0) & \text{si } p^0 \neq 0 \text{ et } -t_f \leq t_b \leq 0, \\ -t_f \operatorname{signe}(p^0) & \text{si } p^0 \neq 0 \text{ et } t_b < -t_f. \end{cases} \quad (4.18)$$

$$x_f^0(p^0, p_f) = \begin{cases} \frac{1}{2}t_f^2 \operatorname{signe}(p_f) & \text{si } p^0 = 0, \\ \frac{1}{2}t_f^2 \operatorname{signe}(p^0) & \text{si } p^0 \neq 0 \text{ et } t_b > 0, \\ (\frac{1}{2}t_f^2 - t_b^2) \operatorname{signe}(p^0) & \text{si } p^0 \neq 0 \text{ et } -t_f \leq t_b \leq 0, \\ -\frac{1}{2}t_f^2 \operatorname{signe}(p^0) & \text{si } p^0 \neq 0 \text{ et } t_b < -t_f, \end{cases} \quad (4.19)$$

**Remarque.** On notera que les fonctions  $x_f(\cdot)$  et  $x_f^0(\cdot)$  ne sont pas obtenues en utilisant les équations (2.19) et (2.20) mais en exprimant d'abord le cône normal  $N_{\mathcal{A}}(\hat{x}_f)$  pour tous les points  $\hat{x}_f = (x_f^0, x_f) \in \partial\mathcal{A}$ , puis en donnant l'expression de  $x_f$  et  $x_f^0$  en fonction de  $\hat{p}_f \in N_{\mathcal{A}}(\hat{x}_f)$ . Par simplicité, ces calculs ne sont pas décrits, mais la démarche reste importante.

Comme nous pouvons le voir dans la figure 4.7a, nous avons construit une ellipse  $\mathcal{E}$  qui approche  $\mathcal{A}$ . Nous construisons le difféomorphisme linéaire  $\phi$  qui transforme cette ellipse en cercle, comme nous l'avons fait dans la section 4.3.2. La transformation de  $\mathcal{A}$  par  $\phi$  est montrée dans la figure 4.7b.

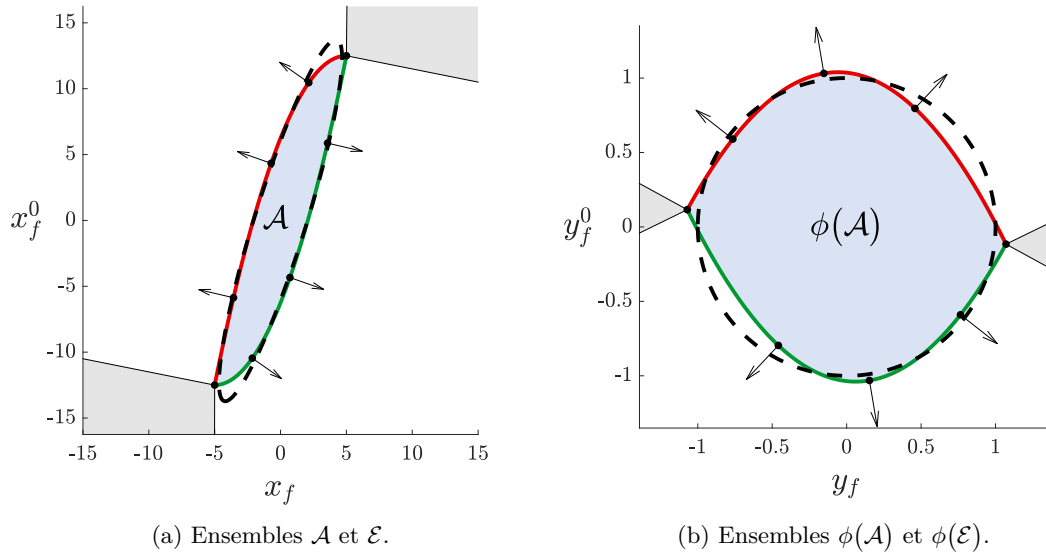


FIGURE 4.7 – Transformation de l'ensemble accessible  $\mathcal{A}$  et son approximation  $\mathcal{E}$  par  $\phi$ , avec  $t_f = 5$  et  $x_0 = 0$ .

Les fonctions de tir  $T_1$  et  $T_2$  dans les nouvelles coordonnées sont construites à partir des équations (4.16) et (4.17), et sont affichées dans la figure 4.8.

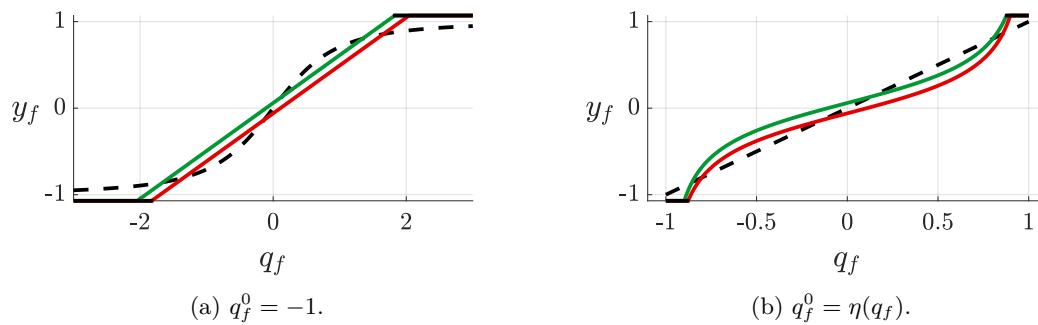


FIGURE 4.8 – Fonction de tir  $T_1$  (à gauche) et  $T_2$  (à droite) avec  $t_f = 5$  et  $x_T = y_T = 0$ .

#### 4.3.4 Cas général : utilisation de l'application exponentielle

On s'intéresse toujours à la résolution du problème

$$(O_x) \quad \begin{cases} \min_{\hat{x}_f} x_f^0 \\ \text{s.c. } \hat{x}_f \in \mathcal{A}, \\ x_f = x_T, \end{cases}$$

mais cette fois en supposant que le seul moyen d'obtenir un point  $\hat{x}_f \in \partial\mathcal{A}$  est d'évaluer l'application exponentielle  $\exp_{\vec{h}}(\hat{p}_0)$  en un co-état augmenté initial  $\hat{p}_0$  non trivial.

**Remarque.** On supposera dans la suite que l'image  $(\hat{x}_f, \hat{p}_f)$  de  $\hat{p}_0$  par l'application exponentielle  $\exp_{\vec{h}}$  appartient à  $\partial\mathcal{A} \times N_{\mathcal{A}}(\hat{x}_f)$ , ce que nous constatons empiriquement dans les applications numériques présentées ici.

##### 4.3.4.1 Méthode de résolution classique

Numériquement, on cherche donc un co-état initial  $\hat{p}_0 = (p_0^0, p_0)$  tel que

$$\hat{p}_0 \neq 0, \quad p_0^0 \leq 0, \quad \pi_x(\exp_{\vec{h}}(\hat{p}_0)) = x_T,$$

ce qui peut se faire en cherchant numériquement un zéro non trivial de la fonction de tir  $S: \mathbb{R}^- \times \mathbb{R}^n \rightarrow \mathbb{R}^n$  définie par

$$S(\hat{p}_0) = \pi_x(\exp_{\vec{h}}(\hat{p}_0)) - x_T.$$

Cette méthode correspond directement à la méthode de tir simple, présenté dans la section 2.2.1.

##### 4.3.4.2 Méthode de tir avec preconditionnement

Comme dans le cas précédent, nous considérons toujours que l'on sait construire des points de la frontière de l'ensemble accessible  $\mathcal{A}$ . Cependant, nous nous affranchissons maintenant de la seconde hypothèse du Cas 2, à savoir que l'on ne connaît pas la fonction  $\hat{x}_f(\cdot)$ . Pour pouvoir nous appuyer sur les développements précédents schématiquement représentés en figure 4.6, nous allons donc devoir la construire. On connaît pour cela une relation naturelle entre  $\hat{p}_0$  et  $\hat{x}_f$  :

$$\hat{x}_f(\hat{p}_0) = \pi_{\hat{x}}(\exp_{\vec{h}}(\hat{p}_0)),$$

mais il nous manque un lien entre  $\hat{p}_f$  et  $\hat{p}_0$ . Pour établir ce lien, on suppose que la fonction

$$\hat{p}_0 \mapsto \hat{p}_f(\hat{p}_0) = \pi_{\hat{p}}(\exp_{\vec{h}}(\hat{p}_0))$$

est inversible et on note  $\hat{p}_0(\cdot)$  sa réciproque. La fonction  $\hat{x}_f(\cdot)$  est alors donnée par

$$\hat{x}_f(\hat{p}_f) = \pi_{\hat{x}}(\exp_{\vec{h}}(\hat{p}_0(\hat{p}_f))).$$

Cependant, la connaissance de la fonction  $\hat{p}_0(\cdot)$  est une hypothèse assez restrictive. On propose plutôt d'utiliser une approximation  $\hat{\varphi}_0(\cdot)$  de la fonction  $\hat{p}_0(\cdot)$  et notre fonction  $\hat{x}_f(\cdot)$  est alors approximativement

$$\hat{x}_f(\hat{p}_f) \approx \pi_{\hat{x}} \left( \exp_{\vec{h}}^{\rightarrow} (\hat{\varphi}_0(\hat{p}_f)) \right).$$

En utilisant le développement de la section 4.3.3.2, la fonction  $T: \mathbb{R}^{n+1} \rightarrow \mathbb{R}^n$  est définie par

$$\begin{aligned} T(\hat{q}_f) &= (\pi_y \circ \phi \circ \pi_{\hat{x}} \circ \exp_{\vec{h}}^{\rightarrow} \circ \hat{\varphi}_0) (J_{\phi^{-1}}(\hat{y}_f)^{-\top} \hat{q}_f) - y_T \\ &= A_x \pi_x (\exp_{\vec{h}}^{\rightarrow} \circ \hat{\varphi}_0) (A^\top \hat{q}_f) + b_x - y_T \end{aligned}$$

et peut être schématisée comme en figure 4.9 à comparer au schéma de la figure 4.6. Les fonctions  $T_1$  et  $T_2$  sont construites de la même manière que précédemment.

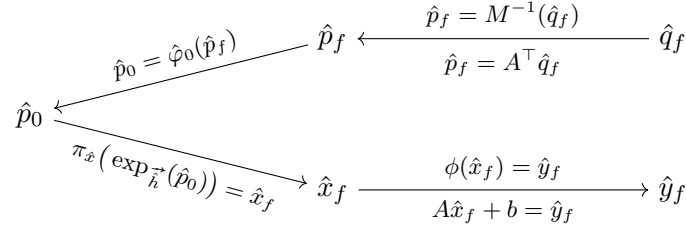


FIGURE 4.9 – Représentation schématique de la fonction de tir avec preconditionnement sous les hypothèses du cas général.

On remarque que si  $\hat{\varphi}_0(\cdot) = \hat{p}_0(\cdot)$  et si  $\phi(\mathcal{A}) = \mathcal{B}^{n+1}$ , alors les fonctions  $T_1$  et  $T_2$  correspondent à celles de la section 4.3.2. Plus précisément, la fonction  $T_2$  serait une fonction affine définie sur  $\mathcal{B}^{n+1}$ , ce qui permettrait la convergence d'un solveur de type Newton une seule itération à partir d'une condition initiale  $p_f \in \mathcal{B}^{n+1}$ . De plus, un zéro de  $T_2$  serait donné par  $p_f = y_T$ .

Ces deux hypothèses sont très restrictives : dans une application industrielle, il est en effet très peu probable que l'on puisse connaître  $\hat{p}_0(\cdot)$  et un tel difféomorphisme  $\phi(\cdot)$ . Cependant, en considérant que  $\hat{\varphi}_0(\cdot) \approx \hat{p}_0(\cdot)$  et que  $\phi(\mathcal{A}) \approx \mathcal{B}^{n+1}$ , on est en droit d'espérer que la fonction de tir  $T_2$  se rapproche de la fonction linéaire, ce qui permettrait à notre solveur de converger plus rapidement. De plus, quand bien même le point  $y_T$  ne serait plus un zéro de  $T_2$ , il pourrait néanmoins constituer une bonne initialisation pour notre solveur.

#### 4.3.4.3 Exemple

Pour illustrer la méthode de preconditionnement proposée dans ce cas général, nous allons l'appliquer au problème (E). À l'inverse de ce que nous avons fait dans la section 4.3.3.3, nous n'avons pas besoin de construire l'ensemble accessible augmenté  $\mathcal{A}$ , d'avoir une expression des cônes normaux pour tout  $\hat{x}_f \in \partial\mathcal{A}$  et d'en déduire les fonctions  $x_f^0$  et  $x_f$  donnée par les équations (4.19) et (4.18).

En s'appuyant sur les développements de la page 29, les fonctions  $\hat{p}_f(\cdot)$  et  $\hat{p}_0(\cdot)$  sont données par

$$\hat{p}_f(p^0, p_0) = (p^0, p_0 - p^0 t_f) \quad \text{et} \quad \hat{p}_0(p^0, p_f) = (p^0, p_f + p^0 t_f).$$

En calculant la fonction

$$\hat{x}_f(\hat{p}_f) = \pi_{\hat{x}} \left( \exp_{\vec{h}} \left( \hat{p}_0(\hat{p}_f) \right) \right),$$

et en remarquant que  $t_a = t_b + t_f$ , avec  $t_a = \frac{p_0}{p^0}$  et  $t_b = \frac{p_f}{p^0}$ , nous retombons directement sur les fonctions (4.19) et (4.18). Les fonctions de tir  $T_1$  et  $T_2$  correspondent donc aux fonctions construites dans la section 4.3.3.3, c'est-à-dire celles qui sont affichées dans la figure 4.8. Comme prévu, la fonction de tir  $T_2$  est quasiment affine, et on constate de plus que la zone utile est étendue (elle occupe presque tout le segment  $[-1, 1]$  dans le cas de la fonction  $T_2$ ).

## Conclusion

Nous venons de proposer une méthode de preconditionnement de la fonction de tir basée sur deux développements théoriques, à savoir l'interprétation géométrique du co-état augmenté effectué dans la section 4.2 dans le cas où l'ensemble accessible augmenté est convexe fermé, et la transformée de Mathieu qui fournit le relèvement dans l'espace des phases à partir d'un difféomorphisme sur l'espace d'état, détaillé dans l'annexe A.

Plus précisément, à partir d'une ellipse  $\mathcal{E}^{n+1}$  qui approche l'ensemble accessible augmenté  $\mathcal{A}$ , on détermine un bon difféomorphisme linéaire  $\phi$  qui transforme cette ellipse en la boule unité  $\mathcal{B}^{n+1}$ . À partir de ce difféomorphisme et d'une fonction  $\hat{\varphi}_0$  qui nous fournit une correspondance approchée entre le co-état final  $\hat{p}_f$  et le co-état initial  $\hat{p}_0$ , il est possible de construire la fonction de tir  $T$  schématisée dans la figure 4.9, puis d'en déduire les deux fonctions de tir  $T_1$  et  $T_2$ .

L'idée principale qui nous laisse penser que la convergence du solveur peut être plus rapide avec la fonction  $T_2$  qu'avec les fonctions de tir  $S_1$  ou  $S_2$  dans les coordonnées initiales est le fait que si l'ensemble accessible est l'ellipse  $\mathcal{E}^{n+1}$  et si  $\hat{\varphi}_0$  correspond à la vraie correspondance entre le co-état augmenté final  $\hat{p}_f$  et initial  $\hat{p}_0$ , alors la fonction  $T_2$  est une fonction affine définie sur la boule  $\mathcal{B}^n$ . Ainsi, si l'approximation de  $\mathcal{A}$  par  $\mathcal{E}^{n+1}$  et de  $\hat{p}_0(\cdot)$  par  $\hat{\varphi}_0(\cdot)$  est suffisamment bonne, on peut espérer que la fonction de tir  $T_2$  soit proche d'une fonction affine et donc soit bien adaptée à un solveur de type Newton.

Nous avons appliqué cette méthode sur le problème (E), et nous avons obtenu une fonction de tir  $T_2$  assez proche d'une fonction affine comme nous pouvons le constater dans la figure 4.8. Dans cet exemple simple, il n'est bien sûr pas nécessaire d'utiliser la méthode proposée pour obtenir une solution du problème : la méthode de tir simple fournit déjà une fonction de tir  $T_1$  linéaire par morceaux, comme nous l'avons vu dans la section 2.24, dont la recherche de zéro peut être faite de manière analytique et ne nécessite donc pas l'utilisation d'un solveur numérique. Dans le prochain chapitre, cette méthode de preconditionnement géométrique sera appliquée au problème de répartition de couple, ce qui permettra de montrer sa pertinence dans une application industrielle plus complexe.



## Expérimentations numériques

---

<b>5.1</b>	<b>Méthode de tir simple</b> .....	<b>94</b>
5.1.1	Flot hamiltonien .....	94
5.1.2	Fonctions de tir .....	97
5.1.3	Méthode numérique .....	99
<b>5.2</b>	<b>Méthode à deux niveaux</b> .....	<b>100</b>
5.2.1	Construction d'approximations des fonctions valeur .....	101
5.2.2	Résolution .....	106
5.2.3	Résultats .....	108
5.2.4	Avantages .....	112
<b>5.3</b>	<b>Préconditionnement géométrique</b> .....	<b>112</b>
5.3.1	Preuve de concept .....	113
5.3.2	Généralisation .....	118

Dans les chapitres précédents, nous avons présenté les méthodes numériques indirectes classiques dans le chapitre 2 et développé deux nouvelles méthodes : la méthode Macro-Micro basée sur la décomposition à deux niveaux du problème (OCP) dans le chapitre 3, et la méthode de préconditionnement géométrique de la fonction de tir dans le chapitre 4, basée sur l'interprétation géométrique du co-état et sur la transformée de Mathieu.

L'objectif de ce chapitre est d'appliquer ces méthodes au problème de répartition de couple dans les véhicules hybrides présenté dans le chapitre 1. Les résultats associés à la méthode Macro-Micro peuvent être trouvés dans [21]. On propose également une méthode efficace afin d'obtenir les bases de données des fonctions valeur pour la méthode Macro-Micro dans la section 5.2.1.1. Cette méthode et les résultats associés correspondent à ceux du proceeding [22].

On propose d'abord de construire l'application exponentielle du flot hamiltonien et d'appliquer la méthode de tir simple dans la section 5.1. Nous allons ensuite appliquer la méthode Macro-Micro dans la section 5.2, et comparer cette méthode en termes de temps de calcul, de nombre de calculs et d'optimalité à la méthode de tir simple et à une autre méthode, la méthode *Empirique*, qui permettra de montrer la pertinence du problème (Macro). Finalement, nous allons appliquer la méthode de préconditionnement géométrique au premier problème de commande optimale du problème (Micro) dans la section 5.3 et montrer que cette méthode permet de réduire le nombre d'itérations de notre solveur.



## 5.1 Méthode de tir simple

Introduit dans le chapitre 1, le problème de répartition du couple dans notre véhicule hybride ( $P_3$ ) est de la même forme que le problème (OCP). Afin d'utiliser les notations introduites dans les trois chapitres précédents, on propose de considérer dans ce chapitre que le problème (OCP) correspond directement au problème ( $P_3$ ), c'est-à-dire

$$(OCP) \quad \begin{cases} \min_{x,u} \int_{t_0}^{t_f} f^0(t, x(t), u(t)) dt \\ \text{s.c. } \dot{x}(t) = f(t, x(t), u(t)), & t \in [t_0, t_f] \text{ p.p.}, \\ u(t) \in U(t), & t \in [t_0, t_f], \\ x(t_0) = x_0, \quad x(t_f) = x_T, \end{cases}$$

où l'état  $x \in AC([t_0, t_f], \mathbb{R})$  correspond à l'état de charge de la batterie *SOC*, la commande  $u \in L^\infty([t_0, t_f], \mathbb{R}^2)$  correspond à la paire  $(T_{qICE}, Gear)$ . De plus, on considère le cycle connu, et nous avons choisi le WLTC (c.f. Figure 1.2), c'est-à-dire que  $t_0 = 0$  seconde et  $t_f = 1800$  secondes.

**Remarque.** Dans le problème ( $P_3$ ), la fonction  $f^0$  ne dépend pas de l'état  $x$ . On propose cependant de marquer cette dépendance, même si elle n'a pas d'impact sur la dynamique de consommation en carburant, afin de se rapprocher de la formulation classique introduite dans le chapitre 2.

Les fonctions  $f^0$  et  $f$ , ainsi que les fonctions qui composent l'ensemble  $U(t)$  sont données à travers un code en `Matlab Simulink`.

### 5.1.1 Flot hamiltonien

**Calcul de la commande maximisante.** Le pseudo-hamiltonien  $h$  associé au problème (OCP) est défini de manière classique par

$$h(t, x, p, p^0, u) = p^0 f^0(t, x, u) + pf(t, u).$$

Pour le temps  $t$ , l'état  $x$ , le co-état  $p$  et  $p^0$  fixés, la commande  $u^*$  doit maximiser le pseudo-hamiltonien

$$u^* \in \arg \max_{w \in U(t)} h(t, x, p, p^0, w).$$

Du fait de l'utilisation des fonctions tabulées dans notre modèle et du caractère discret d'une partie de la commande (rapport de boîte de vitesse), une méthode simple et naturelle pour chercher le maximum du pseudo-hamiltonien consiste à considérer une discrétisation  $\tilde{U}^k(t)$  de  $U(t)$  en  $k$  points par valeur de *Gear* non nulle et à choisir une commande  $u^*$  qui maximise  $h$  dans cet ensemble discret

$$u^* \in \arg \max_{w \in \tilde{U}^k(t)} h(t, x, p, p^0, w).$$

En utilisant la formulation donnée par l'équation (1.3) de  $U(t)$ , l'ensemble discrétisé  $\tilde{U}^k(t)$  est construit par

$$\tilde{U}^k(t) = \left\{ (T_{qICE}, Gear) \left| \begin{array}{l} Gear \in U_1(t), \\ T_{qICE}(t) \in \tilde{U}_2^k(t, Gear) \end{array} \right. \right\},$$

où  $\tilde{U}_2^k$  correspond à une discrétisation uniforme de  $k$  points de  $U_2$ , définie par

$$\tilde{U}_2^k(t, G) = \{T_-(t, G)\} \cup \bigcup_{i=1}^{k-1} \left\{ T_+^-(t, G) + \frac{i-1}{k-2} (T_+^+(t, G) - T_+^-(t, G)) \right\}.$$

Autrement dit, l'ensemble  $\tilde{U}^k(t)$  est construit à partir de  $k$  points uniformément répartis entre les couples  $T_{qICE}$  minimum et maximum, pour tous les  $Gear$  non nuls possibles. Si tous les  $Gear$  sont admissibles au temps  $t \in [t_0, t_f]$ , l'ensemble  $\tilde{U}^k(t)$  est donc constitué de  $4k + 1$  points : un point pour  $Gear = 0$  et  $k$  points pour les quatre autres  $Gear$ .

On peut voir le choix de la commande maximisante comme un « combat » entre 5 hamiltoniens, qui correspondent respectivement aux hamiltoniens où le  $Gear$  est fixé. En effet, on peut construire pour tout  $Gear \in U_1(t)$  le hamiltonien  $h_{Gear}$  défini par

$$h_{Gear}(T_{qICE}) = h(t, x, p, p^0, (T_{qICE}, Gear))$$

et comparer le maximum de chacune de ces fonctions. Ce point de vue est illustré dans la figure 5.1.

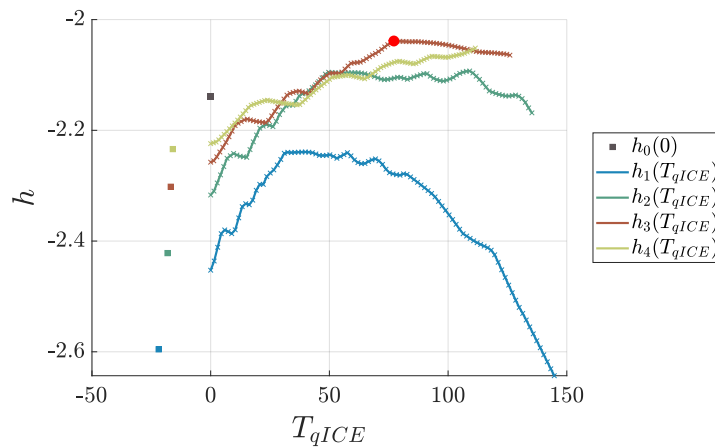


FIGURE 5.1 – Illustration du « combat » entre les hamiltoniens. Dans le cas illustré, le « combat » est remporté par  $h_3$ . La commande maximisante correspond donc au couple  $(T_{qICE}, 3)$ , où  $T_{qICE}$  correspond à l'abscisse du point rouge qui maximise  $h_3$ . Les croix sur les courbes correspondent aux points de  $\tilde{U}_{T_{qICE}}^k(t, Gear)$ , avec  $k = 100$ . Les carrés correspondent au singleton dans la définition de  $\tilde{U}^k$ .

Numériquement, le calcul de  $h(t, x, p, p^0, \tilde{U}^k(t))$  est effectué de manière vectorisée. On en déduit ensuite la commande maximisante  $u^*$  correspondant au maximum de cet ensemble de valeurs.

**Dynamique pseudo-hamiltonienne.** Pour  $t$ ,  $x$ ,  $p^0$  et  $p$  donnés et pour chacune des valeurs discrètes de la commande, nous avons calculé à l'étape ci-dessus les valeurs de  $f$  et de  $f^0$  qui nous ont permis de former le pseudo-hamiltonien et d'en déduire la commande maximisante  $u^*$ . On peut donc récupérer des calculs précédents les valeurs de  $f$  et de  $f^0$  en  $u^*$ , ce qui nous permet de connaître

$$\dot{x}(t) = f(t, x(t), u^*(t)) \quad \text{et} \quad \dot{x}^0(t) = f^0(t, u^*(t)).$$

La dynamique du co-état

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

est quant à elle déterminée par la méthode des différences finies sur  $h$ . L'ensemble de ces calculs est effectué en `Matlab Simulink`.

**Application exponentielle du pseudo-hamiltonien.** L'intégrateur numérique utilisé pour la dynamique hamiltonienne est directement fourni dans `Matlab Simulink`, et correspond à la méthode d'Euler explicite d'ordre 1 à pas fixe. Ce pas est choisi suffisamment petit (100 ms) pour garantir une certaine précision sur la solution. L'intégration de la dynamique pseudo-hamiltonienne est réalisée par la fonction `flot` depuis `Matlab`, qui prend en entrée l'état initial  $x_0$ , le co-état initial  $p_0$ , la valeur de  $p^0$ , le temps initial  $t_0$  et final  $t_f$  ainsi que les deux séries temporelles  $N_w(\cdot)$  et  $T_{qW}(\cdot)$  qui proviennent du cycle, et qui sont considérées fixées ici. Cette fonction nous renvoie la trajectoire d'état  $x(\cdot)$ , de co-état  $p(\cdot)$ , de coût  $x^0(\cdot)$ , ainsi que la commande maximisante  $u^*(\cdot)$ . On a donc

$$(x(\cdot), p(\cdot), u^*(\cdot), x^0(\cdot)) = \text{flot}(x_0, p_0, p^0, t_0, t_f). \quad (5.1)$$

Même si le modèle est vectorisé, l'appel au `flot` est assez coûteux. En effet, pour  $t_0 = 0$  et  $t_f = 1800$ , l'évaluation de `flot`( $x_0, p_0, p^0, t_0, t_f$ ) prend en moyenne 30.2 secondes. Cette moyenne a été calculée à partir de 100 évaluations de la fonction, pour différentes valeurs d'état initial  $x_0$ , de co-état initial  $p_0$ , avec  $p^0 = -1$ . L'application exponentielle  $\exp_{\vec{h}}$  est définie à partir du `flot`, en récupérant les valeurs finales  $x(t_f)$  et  $p(t_f)$

$$\exp_{\vec{h}}(t_f, t_0, x_0, p_0, p^0) = (x(t_f), p(t_f)) \quad \text{avec} \quad (x(\cdot) \text{ et } p(\cdot)) \text{ calculées par (5.1)}.$$

**Remarque.** Le choix du langage de programmation `Matlab Simulink` ainsi que de ce mode de programmation par blocs est justifié par le fait qu'ils sont largement utilisés dans le monde industriel, en particulier pour le modèle utilisé décrit au chapitre 1. Il nous a également paru important de conserver les mêmes méthodes numériques que celles déjà utilisées par l'entreprise (méthode de calcul de la commande maximisante, méthode d'intégration, ...) afin que les méthodes proposées dans ce document puissent être employées et adaptées par d'autres personnes et pour potentiellement d'autres problèmes de commande optimale semblables.

## 5.1.2 Fonctions de tir

La méthode de tir simple décrite dans le chapitre 2 peut être appliquée au problème considéré. À cet effet, on construit la fonction de tir  $S: \mathbb{R} \times \mathbb{R}^- \rightarrow \mathbb{R}$  définie par

$$S(p_0, p^0) = \exp_{\vec{h}}(t_0, t_f, x_0, p_0, p^0) - x_T$$

où le temps initial  $t_0$ , le temps final  $t_f$  ainsi que l'état initial  $x_0$  sont fixés. Les fonctions de tir  $S_1: \mathbb{R} \rightarrow \mathbb{R}$  et  $S_2: [-1, 1] \rightarrow \mathbb{R}$  sont construites à partir des équations (2.22) et (2.23) et sont montrées à travers la figure 5.2.

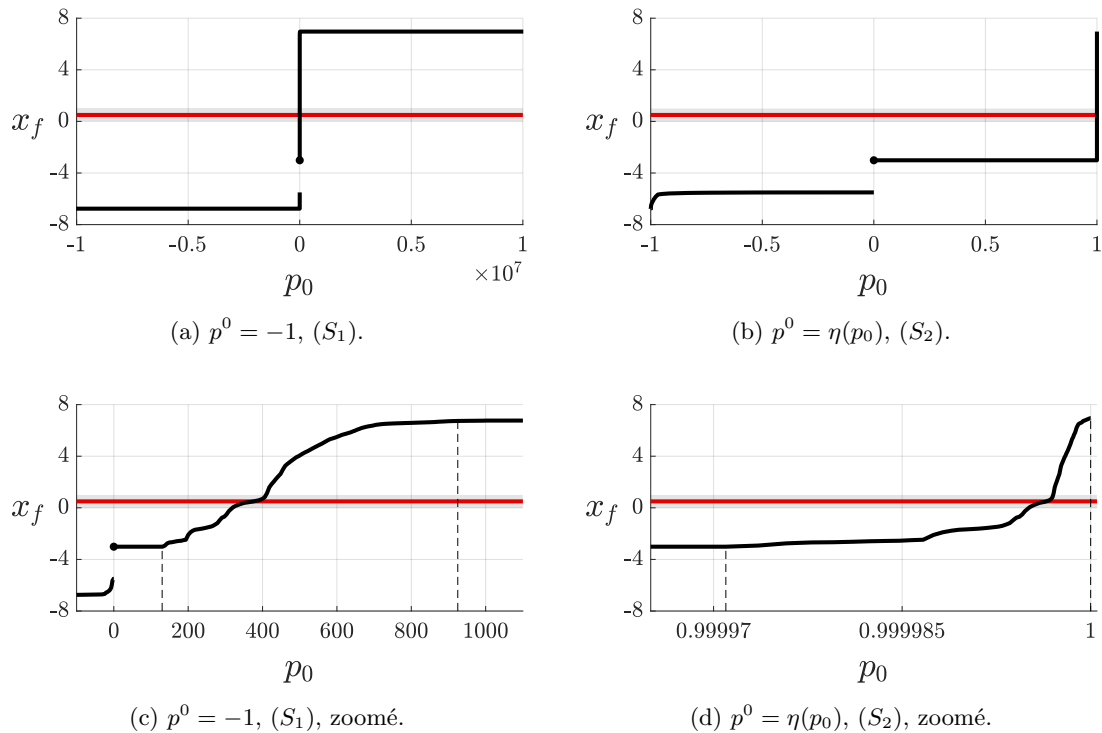


FIGURE 5.2 – Évolution de l'état final  $x_f$  en fonction du co-état initial  $p^0$ . L'état initial  $x_0 = 0.5$  est fixé, tout comme le temps initial  $t_0 = 0$  et le temps final  $t_f = 1800$ . Le multiplicateur  $p^0$  est égal à  $-1$  dans les figures 5.2a et 5.2c, et prend la valeur  $p^0 = \eta(p_0)$  dans les figures 5.2b et 5.2d. La zone grisée  $[0.2, 0.8]$  correspond à l'ensemble des états finaux admissibles. Les figures 5.2c et 5.2d correspondent respectivement à un agrandissement des figures 5.2a et 5.2b sur la zone « intéressante », c'est-à-dire où l'état final  $x_f$  évolue. La ligne rouge correspond à l'hyperplan  $x_f = x_T := 0.5$ . L'objectif de la méthode de tir est de trouver l'intersection entre la fonction en noir et la droite rouge. Les fonctions de tir  $S_1$  et  $S_2$  correspondent respectivement à une translation de  $x_T$  des fonctions des figures 5.2a-5.2c et 5.2b-5.2d.

**État final.** Comme nous pouvons le remarquer, l'état final  $x_f$  peut évoluer dans l'intervalle  $[-6.7, 7]$ , alors que notre modèle n'a pas été développé pour prendre en compte des valeurs hors de l'intervalle  $[0.2, 0.8]$  correspondant à la contrainte d'état donnée par l'équation (1.2). Nous avons permis à notre modèle d'extrapoler sur ces valeurs non physiques, et ce pour une raison principale : nous voulions que l'application exponentielle puisse être bien définie pour tout couple  $(p_0, p^0) \in \mathbb{R}^2$  non trivial, et donc que notre modèle nous fournisse une valeur et non une erreur. De cette façon, les fonctions de tir  $S_1$  et  $S_2$  peuvent être définies sur tout leur intervalle de départ.

**Saut des fonctions de tir.** Comme nous pouvons le constater sur les figures 5.2b-5.2c, les fonctions de tir  $S_1$  et  $S_2$  ne sont pas continues en 0. Cette particularité est due au fait que l'application exponentielle  $\exp_{\vec{h}}$  est mal définie pour  $p_0 = 0$ . En effet, dans ce cas, il existe une infinité de quadruplets  $(x(\cdot), p(\cdot), u(\cdot), p^0)$  qui satisfont le principe du maximum. L'application exponentielle définie sur le domaine de l'hypothèse (H2) n'est donc pas une fonction mais une multi-application.

Pour comprendre physiquement ce phénomène, nous allons d'abord expliquer ce qu'il se passe au niveau du point noir des figures 5.2a-5.2c. La stratégie associée est assez simple : ne jamais utiliser le moteur thermique ( $Gear = 0$  et  $T_{qICE} = 0$ ) (mode électrique pur), et ainsi avoir une consommation en carburant nulle. On pourrait donc penser que cet état de charge de batterie final est le minimum atteignable, cependant, comme nous pouvons le constater, ce n'est pas le cas. Il est en effet possible d'atteindre des états de charge de batterie finaux plus petits en utilisant non plus uniquement le moteur électrique en phase de décélération mais également le moteur thermique (ce qu'on appelle le frein moteur). Ainsi, non seulement la consommation de carburant reste nulle mais en plus la batterie n'est pas rechargée au maximum par récupération de l'énergie de freinage. Ces stratégies sont donc totalement invraisemblables dans le problème considéré, étant donné qu'elles gaspilleraient de l'énergie au lieu de la préserver : l'état final cible ne sera jamais plus petit que celui atteint en mode électrique pur, donné par le point noir.

On considère donc que les extrémales d'intérêt de notre problème sont celles dont le co-état initial est positif. Sur cet ensemble, l'application exponentielle s'avère être bien définie. Afin d'éviter tout problème numérique, nous avons donc imposé à la fonction `flot` de nous fournir les trajectoires associées à la commande  $u(\cdot) = 0$  (point noir) dans le cas où  $p_0 = 0$ .

**Remarque.** Il serait également possible d'enlever ce saut sur les fonctions de tir en ajoutant une pénalité sur l'utilisation du frein moteur.

**A priori.** En comparant respectivement les figures 5.2a et 5.2b aux figures 5.2c et 5.2d, on remarque que pour les deux fonctions de tir  $S_1$  et  $S_2$ , une information supplémentaire doit être fournie au solveur afin d'espérer qu'il puisse converger. Cette information a priori correspond à la zone d'évolution « intéressante » des fonctions de tir, c'est-à-dire la zone en co-état où l'état final évolue, donnée par l'intervalle entre les lignes en pointillé. En dehors de cet intervalle (avec  $p_0 > 0$ ), la fonction de tir est essentiellement constante, ce qui empêcherait un solveur de type Newton de converger vers un zéro.

**Solution.** Comme nous pouvons le constater, les deux fonctions de tir  $S_1$  et  $S_2$  sont monotones. De plus, les extrémales  $(x, p, p^0, u)$  associées aux zones plates de ces fonctions ont la même trajectoire d'état  $x(\cdot)$  et de commande  $u(\cdot)$ . Autrement dit, pour tout état initial  $x_0 \in \mathbb{R}$  et final  $x_f \in \mathbb{R}$ , il existe au maximum une trajectoire d'état  $x(\cdot)$  et de commande  $u(\cdot)$  qui satisfait le principe du maximum. Dans le problème considéré, le principe du maximum est donc une condition nécessaire et suffisante d'optimalité : trouver un zéro de  $S_1$  ou  $S_2$  revient à trouver une solution du problème (OCP).

**Remarque.** L'ensemble des remarques effectuées ici sur tout le cycle du WLTC ( $t_0 = 0$  et  $t_f = 1800$ ) sont également valables sur des sous-cycles du WLTC.

### 5.1.3 Méthode numérique

Maintenant que nous venons d'analyser les deux fonctions de tir  $S_1$  et  $S_2$  à travers les figures 5.2, nous allons décrire comment nous trouvons une BC-extrémale du problème (P<sub>3</sub>). Tout d'abord, historiquement dans l'entreprise, la fonction de tir utilisée est la fonction  $S_1$ , c'est pourquoi nous allons l'utiliser. Un zéro de  $S_1$  est trouvé au moyen de la fonction `fsolve` de `Matlab`, en utilisant l'algorithme par défaut qui est le `trust-region-dogleg`, avec un calcul de jacobienne fait par différences finies. La figure 5.3 est une capture d'écran de la sortie du solveur.

Iteration	Func-count	f(x)	Norm of step	First-order optimality	Trust-region radius
0	2	0.000985233		0.000157	1
1	4	0.000793294	1	0.000156	1
2	6	0.000579629	1	0.000136	1
3	8	0.000336976	1	0.000104	1
4	10	1.10983e-05	2.5	1.81e-05	2.5
5	12	7.25663e-07	0.61195	4.85e-06	6.25
6	14	2.78249e-08	0.149713	9.31e-07	6.25
7	15	2.78249e-08	0.0298969	9.31e-07	6.25
8	17	2.7304e-10	0.00747423	9.3e-08	0.00747
9	18	2.7304e-10	0.00293625	9.3e-08	0.0187
10	19	2.7304e-10	0.000734062	9.3e-08	0.000734
11	20	2.7304e-10	0.000183516	9.3e-08	0.000184
12	21	2.7304e-10	4.58789e-05	9.3e-08	4.59e-05
13	22	2.7304e-10	1.14697e-05	9.3e-08	1.15e-05
14	23	2.7304e-10	2.86743e-06	9.3e-08	2.87e-06
15	24	2.7304e-10	7.16858e-07	9.3e-08	7.17e-07

FIGURE 5.3 – Évolution d'indicateurs en fonction du nombre d'itérations du solveur, pour la fonction de tir  $S_1$  sur tout le WLTC ( $t_0 = 0$  et  $t_f = 1800$ ), avec  $x_0 = x_T = 0.5$  et une initialisation  $p_0 = 370$  assez proche de la solution (c.f. Figure 5.2c). Ces indicateurs sont :

- **Func-count** : Nombre total d'appels à  $S_1$ .
- **f(x)** : Carré de la valeur de la fonction de tir  $S_1$  à l'itéré courant.
- **Norm of step** : Norme euclidienne du pas courant.
- **First-order optimality** : Norme de la condition nécessaire d'optimalité d'ordre 1. Dans notre cas, cette valeur correspond à  $|S_1'(\cdot)|$ , calculée par différences finies.
- **Trust-region radius** : Rayon utilisé pour la région de confiance à l'itération courante.

Plus précisément, un zéro de la fonction de tir  $S_1$  correspond à un point  $p_0$  tel que

$$S_1(p_0) \leq \varepsilon$$

où  $\varepsilon > 0$  correspond à la tolérance considérée pour le problème (OCP), fixée à  $10^{-3}$  en accord avec l'entreprise. On remarque dans la figure 5.3 qu'un zéro de la fonction de tir a été trouvé à partir de la 5ème itération, pour 12 évaluations de  $S_1$ . Le coût en termes de temps de calcul est de 362 secondes, ce qui correspond donc à un peu plus de 6 minutes. La BC-extrémale  $(x, p, -1, u)$ , ainsi que les données temporelles obtenues sont montrées dans la figure C.1 en annexe.

## 5.2 Méthode à deux niveaux

Comme nous venons de le voir, il est possible d'obtenir une BC-extrémale par la méthode de tir simple, et donc une solution du problème (OCP). Le temps de calcul reste malgré tout trop long pour que cette méthode numérique puisse être utilisée à bord du véhicule. Comme discuté dans la section 3.2.5, la méthode Macro-Micro permet de pallier ce problème. En effet, réduire la longueur de l'intervalle d'intégration permet de réduire proportionnellement le temps de calcul de la fonction `flot`. Autrement dit, l'appel à la fonction `flot`( $x_0, p_0, p^0, t_0, t_f$ ) est  $N$  fois plus long que le calcul de `flot`( $x_0, p_0, p^0, t_1, t_2$ ), où

$$t_f - t_0 = N(t_2 - t_1).$$

Ce résultat assez trivial est montré dans la figure 5.4.

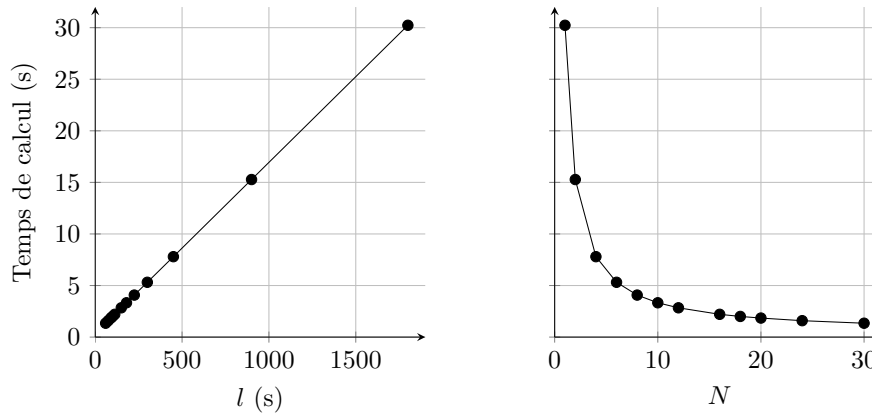


FIGURE 5.4 – Évolution du temps de calcul de la fonction `flot`( $x_0, p_0, p^0, 0, l$ ) en fonction de  $l$  (à gauche) et de la fonction `flot`( $x_0, p_0, p^0, 0, t_f/N$ ) en fonction de  $N$ .

C'est principalement cette idée qui nous a mené au développement de la méthode à deux niveaux. Décrite dans le chapitre 3, elle va être appliquée au problème de répartition de couple. Nous avons choisi arbitrairement de découper l'intervalle de temps initial du WLTC  $[t_0, t_f] = [0, 1800]$  en  $N + 1 = 18$  intervalles de temps  $\Delta_i := [t_i, t_{i+1}]$  de longueur

$$l_i = t_{i+1} - t_i = l = 100 \text{ secondes.}$$

Nous avons donc simplement  $t_i = 100i$ , pour tout  $i \in \mathbb{N}_{N+1}$ , avec  $t_{N+1} = t_f$ . La méthode Macro-Micro est rappelée au travers du schéma de la figure 5.5.

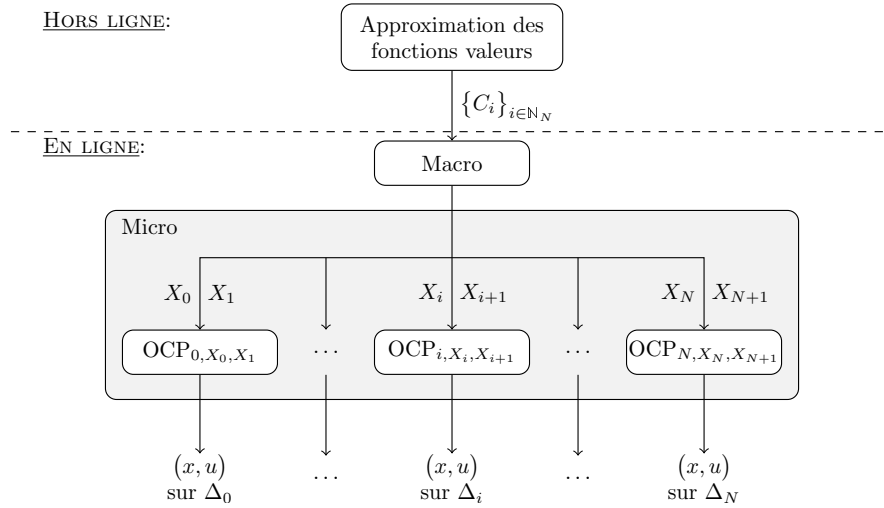


FIGURE 5.5 – Schéma de la méthode Macro-Micro.

Ce schéma met en avant deux caractéristiques principales :

- la méthode Macro-Micro prend en entrée les fonctions  $C_i$ . La création de ces fonctions est une tâche hors ligne ;
- le problème (Micro) est composé de  $N + 1$  problèmes de commande optimale indépendants, définis sur les intervalles de temps  $\Delta_i$  qui sont  $N + 1$  fois plus courts que  $[t_0, t_f]$ .

Dans ce qui suit, nous allons commencer par détailler la construction des fonctions  $C_i$  dans la section 5.2.1. Nous allons ensuite décrire dans la section 5.2.2 les méthodes utilisées pour résoudre le problème (Macro) et les  $N + 1$  problèmes de commande optimale du problème (Micro). Nous discuterons dans la section 5.2.3 des résultats obtenus avec la méthode Macro-Micro et de ses avantages et inconvénients dans la section 5.2.4.

### 5.2.1 Construction d'approximations des fonctions valeur

Pour pouvoir appliquer la méthode à deux niveaux, il faut calculer au préalable, pour tout  $i \in \mathbb{N}_N$ , les approximations  $C_i$  des fonctions valeur  $V_i$  associées au problème (OCP $_{i,a,b}$ ). Dans toute cette section, nous fixons  $i \in \mathbb{N}_N$ . Nous allons d'abord construire une base de données  $\mathbb{D}_i$  d'évaluations de la fonction valeur  $V_i$  associée à l'intervalle  $\Delta_i$  avec une méthode efficace [22], et ensuite utiliser cette base de données afin d'entraîner un réseau de neurones, qui correspondra à la fonction  $C_i$ .

#### 5.2.1.1 Base de données

Nous proposons deux méthodes qui peuvent être utilisées pour créer une base de données  $\mathbb{D}_i$ . La méthode 1 correspond à la méthode naïve qui consiste à évaluer  $V_i$  en certains points admissibles. La méthode 2 est basée sur le calcul d'extrémales.



**Méthode 1.** Cette méthode consiste à évaluer directement la fonction valeur  $V_i$ . On considère donc  $\mathbb{X}_i$  une discrétisation de  $\Omega_i$  (cf. 3.1.1), qui correspond ici à l'ensemble des couples  $(x_i, x_{i+1})$  tels que  $x_{i+1}$  appartient à la zone d'évolution intéressante de l'état final (c.f. section 5.1.2). Pour tout  $(x_i, x_{i+1}) \in \mathbb{X}_i$ , on résout le problème de commande optimale ( $\text{OCP}_{i,x_i,x_{i+1}}$ ) par la méthode de tir simple, et on ajoute à  $\mathbb{D}_i$  le point  $(x_i, x_{i+1}, V_i(x_i, x_{i+1}))$ . Ces étapes sont décrites à travers l'Algorithme 1,

---

**Algorithm 1** Méthode 1

---

**Require:**  $\mathbb{X}_i$

**Ensure:**  $\mathbb{D}_i$

```

for all  $(x_i, x_{i+1}) \in \mathbb{X}_i$  do
   $p_i \leftarrow \text{solve}(S_{i,x_i,x_{i+1}}(p_i) = 0)$ 
   $(x(\cdot), p(\cdot), u(\cdot), x^0(\cdot)) \leftarrow \text{flot}(x_i, p_i, -1, t_i, t_{i+1})$ 
   $\mathbb{D}_i.\text{append}(x(t_i), x(t_{i+1}), x^0(t_{i+1}))$ 
end for

```

---

où la fonction  $S_{i,a,b}$  est définie par

$$S_{i,a,b}(p_0) = \pi_x(\exp_{\vec{h}}(t_{i+1}, t_i, a, p_0, -1)) - b.$$

**Méthode 2.** Cette méthode consiste à remarquer qu'une intégration du `flot`

$$(x(\cdot), p(\cdot), u(\cdot), x^0(\cdot)) = \text{flot}(a, p_0, -1, t_i, t_{i+1})$$

revient à évaluer la fonction valeur  $V_i(a, b)$  en un point  $b$  (inconnu à l'avance) qui sera l'état final  $x(t_{i+1})$ . Notre idée est d'utiliser ce résultat afin de générer une base de données  $\mathbb{D}_i$  à moindre coût. En effet, en considérant  $\mathbb{Z}_i$  une discrétisation de l'espace d'état et co-état initial, on calcule le `flot` pour tout  $(x_i, p_i) \in \mathbb{Z}_i$  et on en déduit le coût ainsi que l'état au temps  $t_{i+1}$ . Cette méthode est décrite dans l'Algorithme 2.

---

**Algorithm 2** Méthode 2

---

**Require:**  $\mathbb{Z}_i$

**Ensure:**  $\mathbb{D}_i$

```

for all  $z_i = (x_i, p_i) \in \mathbb{Z}_i$  do
   $(x(\cdot), p(\cdot), u(\cdot), x^0(\cdot)) \leftarrow \text{flot}(x_i, p_i, -1, t_i, t_{i+1})$ 
   $\mathbb{D}_i.\text{append}(x(t_i), x(t_{i+1}), x^0(t_{i+1}))$ 
end for

```

---

**Comparaisons théoriques.** On se propose maintenant de comparer ces deux méthodes, afin de déterminer celle que l'on va utiliser pour la création de nos bases de données.

**Prérequis :** La méthode 1 nécessite la connaissance de  $\Omega_i$ . En effet, si le couple  $(x_i, x_{i+1}) \notin \Omega_i$ , la fonction de tir  $S_{i,x_i,x_{i+1}}$  n'admet pas de racine, ce qui mènerait donc à une non convergence du solveur. La méthode 2 peut, quant à elle, fonctionner sans connaissance a priori, avec simplement le risque d'obtenir plusieurs fois le même point  $(x_i, x_{i+1}, V_i(x_i, x_{i+1}))$  si on fait appel au `flot` à partir de couples  $(x_i, p_i)$  qui n'appartiennent pas à la zone « intéressante ». Une connaissance, même approximative, de cette zone permettrait donc d'améliorer l'efficacité de cette méthode.

**Temps de calcul :** La méthode 1 est plus coûteuse que la méthode 2. En effet, pour chaque point généré dans la base de données, la méthode 2 nécessite seulement un appel à la fonction `flot` alors que la première méthode met en œuvre une recherche itérative d'un zéro de la fonction de tir, laquelle nécessite au moins un appel à la fonction `flot` par itération (c.f. Figure 5.3). Le nombre moyen d'itérations pour trouver un zéro, évalué sur 475 points  $(x_i, x_{i+1})$  différents à partir de l'initialisation  $p_0 = 500$ , est de 11.2, ce qui implique que pour le même nombre de points, la méthode 1 est au moins 11.2 fois plus coûteuse en termes de temps de calcul que la méthode 2.

**Répartition des données :** la méthode 1 permet d'imposer une répartition donnée de  $\Omega_i$ , par exemple uniforme, ce qui n'est pas possible avec la méthode 2. Cette dernière est donc susceptible de laisser des zones vides. Cependant, nous allons voir plus loin qu'une densité suffisante sur  $\mathbb{Z}_i$  conduit à une distribution satisfaisante dans notre base de données  $\mathbb{D}_i$ .

**Comparaison numérique.** Afin d'évaluer le compromis temps de calcul/répartition des données de la méthode 2, nous allons comparer la distribution des états finaux obtenue par cette méthode à la distribution uniforme choisie pour la méthode 1. Pour ce faire, nous avons considéré le premier intervalle de temps  $\Delta_0 = [t_0, t_1] = [0, 100]$  et nous avons fixé  $x_0 = 0.5$ . Nous avons alors créé la base de données  $\mathbb{D}_0$  avec les deux méthodes, en prenant pour chacune 10 valeurs de  $x_1$  (méthode 1) ou de  $p_0$  (méthode 2) réparties uniformément. Nous avons donc

- l'ensemble  $\mathbb{X}_0$  qui correspond à l'ensemble des couples  $(x_0, x_1)$ , avec  $x_1$  qui appartient à la discrétisation uniforme de 10 points de l'intervalle  $[x_1^-, x_1^+]$ . L'état  $x_1^-$  correspond à l'état au temps  $t_1$  obtenu à partir de la commande  $(0, 0)$  sur tout l'intervalle de temps  $\Delta_0$ . L'état  $x_1^+$  correspond à l'état au temps  $t_1$  obtenu à partir de la commande  $u(\cdot)$  qui maximise pour tout temps. En reprenant la définition du pseudo-hamiltonien, on remarque que cette trajectoire dont la commande  $u(t)$  maximise  $f(t, x(t), \cdot)$  pour tout temps  $t \in [t_0, t_f]$  est associée à une extrémale anormale, avec  $p^0 = 0$  et  $p(t_0) > 0$ .
- l'ensemble  $\mathbb{Z}_0$  qui correspond à l'ensemble des couples  $(x_0, p_0)$ , avec  $p_0$  qui appartient à la discrétisation uniforme de 10 points de l'intervalle  $[p_0^-, p_0^+]$ . Ces deux bornes correspondent aux limites de la zone « intéressante » d'évolution de la fonction de tir  $S_{1,x_0,\cdot}$  et ont été déterminées numériquement.

De plus, nous voulons comparer les deux méthodes en considérant le même temps de calcul. C'est pourquoi nous avons également créé la base de données  $\mathbb{D}_0$  avec la méthode 2 en considérant 112 points, ce qui, en termes de temps de calculs, est en moyenne au

plus aussi coûteux qu'avec les 10 points de la méthode 1 (c.f. paragraphe précédent). La figure 5.6 montre la répartition en état final associée à ces trois expériences.

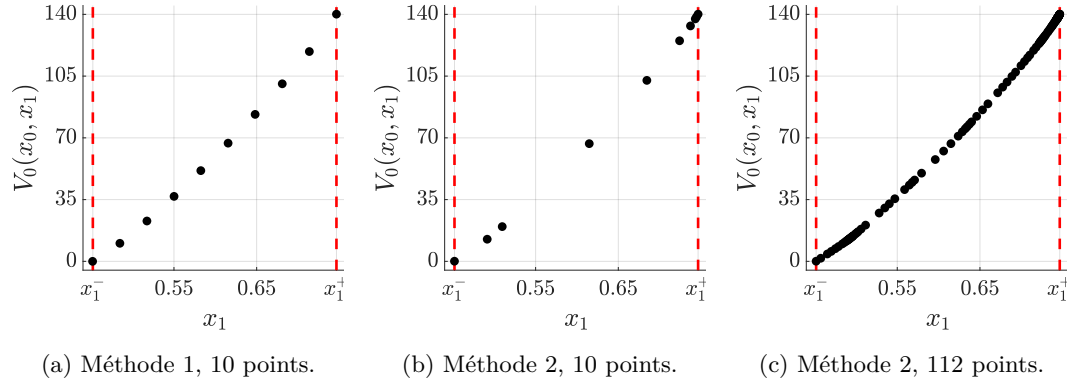


FIGURE 5.6 – Points de la fonction valeur  $V_0$  obtenus avec les deux méthodes.

En comparant les figures 5.6a et 5.6b, on remarque que la répartition obtenue avec la méthode 2 n'est pas uniforme. En comparant les figures 5.6a et 5.6c, on remarque que l'on obtient une meilleure discrétisation de la fonction valeur avec la méthode 2 plutôt qu'avec la méthode 1, et surtout beaucoup plus de points, pour un temps en moyenne au moins équivalent. Principalement pour cette raison, nous avons donc choisi la méthode 2 afin de générer pour tout  $i \in \mathbb{N}_N$  les bases de données  $\mathbb{D}_i$ .

**Choix de la discrétisation pour la méthode 2.** Pour tous les sous-intervalles de temps (pour tout  $i \in \mathbb{N}_N$ ), nous avons donc créé l'ensemble  $\mathbb{Z}_i$  au temps  $t_i$  en discrétisant l'intervalle d'état  $[0.2, 0.8]$  avec un pas de 2.5% et l'intervalle de co-état  $[0, 1000]$  avec un pas de 20. L'intervalle (assez grand) et la discrétisation (assez fine) choisis en co-état nous assure numériquement d'atteindre les bornes de l'ensemble des états finaux accessibles, et également de ne pas avoir de zone vide. Les points en état et co-état de la base de données  $\mathbb{D}_0$  sont montrés dans la figure 5.7.

Pour chacun des 18 intervalles de temps  $\Delta_i$ , il nous faut donc calculer 1275 points, ce qui correspond à 1275 appels à la fonction `flot`. Au total, il nous faut par conséquent 22 950 appels à la fonction `flot` sur des intervalles de temps de 100 secondes. Ces appels prennent en moyenne 1.7 seconde, ce qui veut dire que la création de toutes ces bases de données prendrait au moins 11 heures pour un calcul séquentiel. C'est pourquoi nous avons utilisé la `Matlab Parallel Computing Toolbox` ainsi qu'un ordinateur plus puissant avec `Amazon Web Services` afin de réduire grandement ce temps de calcul.

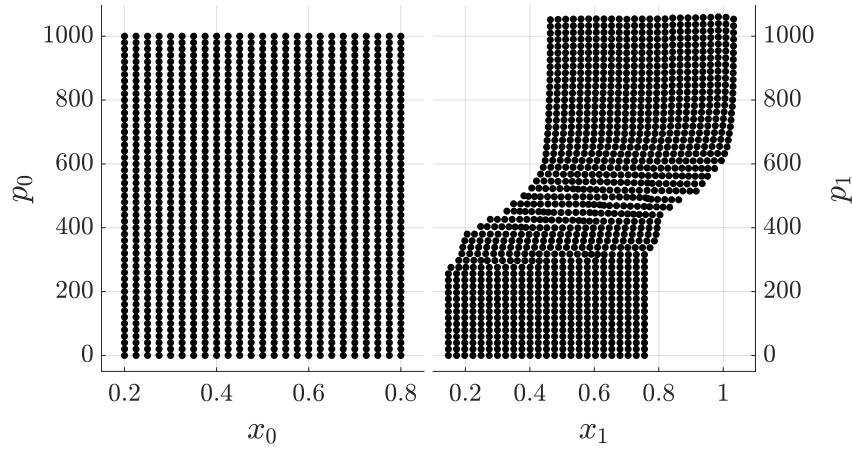


FIGURE 5.7 – Transformation des points  $(x_0, p_0)$  de l'ensemble  $\mathbb{Z}_0$  (à gauche) par le flot hamiltonien, pour fournir l'état et le co-état  $(x_1, p_1)$  au temps  $t_1$ . Le coût optimal associé à chacun de ces points est également calculé afin de construire la base de données  $\mathbb{D}_0$ .

### 5.2.1.2 Approximation des fonctions valeur

L'objectif est maintenant de modéliser les fonctions  $C_i$  qui doivent approcher les fonctions valeur  $V_i$ . On cherche une méthode :

- qui puisse approcher assez finement les points de notre base de données,
- dont le calcul du gradient puisse être fait de manière peu coûteuse,
- qui peut accepter des données non structurées.

Principalement pour ces trois raisons, nous avons choisi de modéliser les fonctions  $C_i$  par des réseaux de neurones simples, de type perceptron multicouche, dont les fonctions d'activation, le nombre de couches, de neurones et de paramètres sont donnés dans la table 5.1.

Couche	Fonction d'activation	Nombre de neurones	Nombre de paramètres
Entrée	id	2	0
Couche cachée 1	tanh	16	48
Couche cachée 2	sigmoïd	8	136
Sortie	id	1	8
Total			192

TABLE 5.1 – Description des réseaux de neurones utilisés pour modéliser les fonctions  $C_i$ .

Nous avons choisi de coder ces réseaux à l'aide du package `TensorFlow` de Python. Classiquement, ils sont entraînés en utilisant l'algorithme `Adam` avec un taux d'apprentissage de 0.01. Après plusieurs comparaisons numériques, nous avons choisi de prendre 4000 epoch avec un `batch size` de 16. Les bases de données  $\mathbb{D}_i$  sont divisées aléatoirement en une base d'entraînement et une base de test avec une répartition de 80% / 20%. Les poids sélectionnés sont ceux qui minimisent la fonction de perte, ici l'erreur quadratique

moyenne, sur la base de test. La figure 5.8 montre le réseau  $C_0$  entraîné ainsi que la base de données  $\mathbb{D}_0$ .

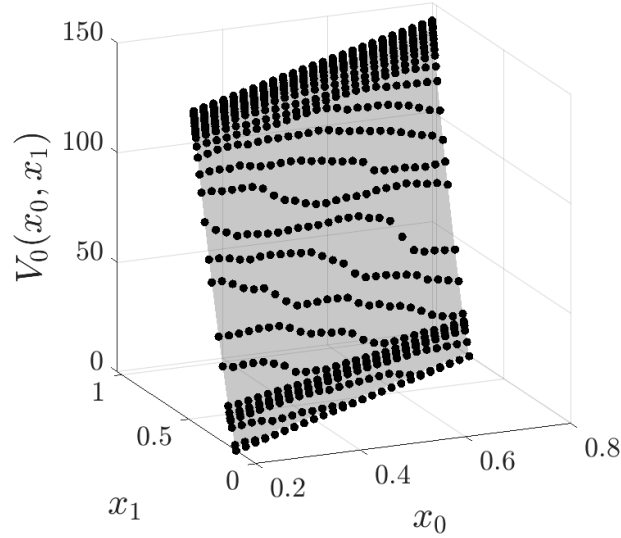


FIGURE 5.8 – Évolution du coût optimal  $V_0(x_0, x_1)$  (points noirs, constituant la base de données  $\mathbb{D}_0$ ) et approché  $C_0(x_0, x_1)$  (nappe grise) en fonction de l'état initial  $x_0$  et final  $x_1$ .

## 5.2.2 Résolution

L'objectif est maintenant de décrire les méthodes utilisées pour la résolution du problème (Macro) et des problèmes de commande optimale du problème (Micro).

### 5.2.2.1 Partie Macro

On rappelle que le problème (Macro) correspond au problème d'optimisation suivant

$$(Macro) \quad \begin{cases} \min_X \sum_{i=0}^N C_i(X_i, X_{i+1}), \\ \text{s.c. } X \in \mathcal{X}, \quad c(X_0, X_{N+1}) = 0, \end{cases}$$

où les fonctions  $C_i$  sont les réseaux de neurones construits et entraînés pour approcher les fonctions valeur. La fonction  $c: \mathbb{R}^2 \rightarrow \mathbb{R}^2$  est définie ici par

$$c(a, b) = (a - x_0, b - x_T).$$

Puisque les fonctions  $C_i$  ne sont que des approximations des fonctions valeur  $V_i$ , il est possible que le problème (Macro) n'ait pas de solution. En fait, dans toutes les expériences considérées, nous avons cherché numériquement  $\tilde{X} = (X_1, \dots, X_N) \in \mathbb{R}^N$  solution du problème sans contraintes suivant

$$\min_X C_0(x_0, X_1) + \sum_{i=0}^{N-1} C_i(X_i, X_{i+1}) + C_N(X_N, x_T), \quad (5.2)$$

et nous avons constaté a posteriori que la solution numérique  $\tilde{X}$  obtenue est admissible pour le problème (Macro), dans le sens où  $(x_0, \tilde{X}, x_T) \in \mathcal{X}$ . Nous nous intéressons donc maintenant à la résolution numérique du problème (5.2).

**Remarque.** L’admissibilité de  $\tilde{X}$  dépend de l’erreur de prédiction des réseaux de neurones  $C_i$  sur leurs domaines d’apprentissage, et également de l’extrapolation (non contrôlée) de ces réseaux de neurones sur les domaines de prédiction où les fonctions valeur ne sont pas définies, c’est-à-dire où il n’existe pas de points dans les bases de données  $\mathbb{D}_i$ . Par « chance », nous n’avons pas eu besoin de prendre en compte l’ensemble  $\mathcal{X}$  dans nos expériences. Dans le cas où nous n’aurions pas eu cette chance, une méthode de pénalisation est proposée dans l’annexe C et pourrait être utilisées dans notre application.

Le problème (5.2) est un problème d’optimisation sans contrainte qui n’a aucune raison apparente d’être convexe. Nous avons comparé plusieurs algorithmes d’optimisation afin d’en sélectionner un qui soit rapide, performant et robuste. Les résultats de ces méthodes d’optimisation sont donnés dans la table 5.2.

Méthode	Temps Moyen	Erreur Moyenne	Variance Moyenne
NLopt MMA	<b>0.7969</b>	50.6417	71.7876
<b>NLopt SLSQP</b>	<b>1.1868</b>	<b>1.6964</b>	2.0579
NLopt DIRECT	19.9442	<b>9.8656</b>	0.0000
NLopt DIRECT-L	26.3860	<b>9.4500</b>	0.0000
NLopt CRS2-LM	98.5034	<b>0.3141</b>	0.0513
SP Nelder-Mead	11.2706	119.4900	682.8640
SP Powell	27.0352	38.2458	69.5863
SP CG	<b>0.6194</b>	36.1308	79.6900
SP BFGS	<b>0.4866</b>	24.7611	874.0601
SP Newton-CG	<b>0.0129</b>	272.7695	908.9777
SP L-BFGS-B	<b>0.3518</b>	16.5693	21.7841
SP TNC	<b>0.5848</b>	67.9273	190.4108
SP COBYLA	<b>2.5188</b>	63.7068	643.8506
<b>SP SLSQP</b>	<b>1.0265</b>	<b>1.4497</b>	1.1479
<b>SP trust-constr</b>	<b>7.2148</b>	<b>0.7955</b>	0.6033

TABLE 5.2 – Comparaison exhaustive de méthodes d’optimisation provenant de deux packages différents de Python : NLopt et SciPy (SP). Ces résultats ont été obtenus à partir de moyennes sur 5 scénarios différents (différentes conditions initiale et finale). Les résultats associés à chacun de ces scénarios sont donnés dans la table C.3 en annexe. Sont mis en gras les temps courts ( $< 10s$ ), les résultats assez précis (erreur moyenne  $< 10g$ ), ainsi que les méthodes qui possèdent ces deux caractéristiques.

Deux méthodes possèdent ces trois caractéristiques : temps court, erreur moyenne faible et petite variance. Il s’agit des méthodes SLSQP et trust-constr. Le choix entre ces deux méthodes est assez difficile et dépend du compromis souhaité entre précision et rapidité. Nous avons choisi de privilégier la rapidité de calcul, c’est pourquoi nous avons utilisé la méthode SLSQP du package SciPy pour résoudre le problème (Macro).

Par ailleurs, pour prendre en compte l'éventuelle non convexité du problème, nous utilisons toujours cet algorithme avec 10 initialisations différentes. La valeur de  $C(\hat{X})$  est donnée dans la table 5.3 pour les cinq scénarios considérés, où  $\hat{X}$  correspond à la meilleure des 10 solutions obtenues. Les faibles valeurs observées dans la colonne « Écart » dans la table 5.3 confirment la pertinence du choix de la méthode SLSQP.

$x_0$	$x_f$	$C(\hat{X})$	Écart
0.3	0.7	915.0890	0
0.4	0.6	837.3952	0,1470
0.5	0.5	767.0753	0.7101
0.6	0.4	696.9034	0.6751
0.7	0.3	627.8010	0.6217

TABLE 5.3 – Coût du problème (Macro) pour les 5 scénarios considérés. La colonne « Écart » correspond à l'écart entre  $C(\hat{X})$  et le coût minimal trouvé parmi toutes les méthodes testées (c.f. Table C.3 en annexe).

### 5.2.2.2 Partie Micro

Maintenant que nous avons obtenu le vecteur d'états intermédiaires  $\hat{X}$ , solution numérique du problème (Macro), nous devons résoudre le problème (Micro), c'est-à-dire les problèmes (OCP $_{i,\hat{X}_i,\hat{X}_{i+1}}$ ) qui s'écrivent de la manière suivante

$$\left\{ \begin{array}{ll} \min_{x,u} \int_{t_i}^{t_{i+1}} f^0(t, x(t), u(t)) dt & \\ \text{s.c. } \dot{x}(t) = f(t, x(t), u(t)), & t \in \Delta_i \text{ p.p.,} \\ u(t) \in U(t), & t \in \Delta_i, \\ x(t_i) = \hat{X}_i, \quad x(t_{i+1}) = \hat{X}_{i+1}. & \end{array} \right.$$

Ces problèmes de commande optimale sont résolus en cherchant un zéro des fonctions de tir simple  $S_{i,\hat{X}_i,\hat{X}_{i+1}}(\cdot)$ , en utilisant les mêmes méthodes numériques que celles décrites pour le tir simple dans la section 2.2.1. Une initialisation naturelle du tir est donnée par le théorème 3.1.5 et s'écrit dans notre cas

$$p_0 = -\nabla_a C_i(\hat{X}_i, \hat{X}_{i+1}). \quad (5.3)$$

Numériquement, le gradient de ces réseaux de neurones est obtenu simplement par rétro-propagation directement codée dans le package **Tensorflow**.

### 5.2.3 Résultats

**Méthode Empirique.** Afin de mettre en évidence l'apport du problème (Macro), nous avons créé une autre méthode, appelée méthode *Empirique* dans laquelle les états intermédiaires  $\bar{X} = (\bar{X}_0, \dots, \bar{X}_{N+1})$  sont linéairement répartis entre  $x_0$  et  $x_T$  : pour tout  $i \in \mathbb{N}_{N+1}$

$$\bar{X}_i = x_0 + ik, \quad \text{où } k = \frac{x_T - x_0}{N + 1}. \quad (5.4)$$

**Remarque.** Dans le cas où le vecteur d'états intermédiaires obtenu n'est pas admissible ( $\bar{X} \notin \mathcal{X}$ ), il est possible de projeter  $\bar{X}$  sur  $\mathcal{X}$ .

**Remarque.** La méthode Empirique peut être vue comme un cas spécifique de la méthode Macro-Micro. En effet, si on pose

$$C_i(a, b) = (b - a)^2, \quad \forall i \in \mathbb{N}_N, \quad (5.5)$$

le vecteur d'états intermédiaires  $\bar{X}$  construit par la méthode Empirique est la solution du problème

$$\begin{cases} \min_X C(X) = \sum_{i=0}^N C_i(X_i, X_{i+1}), \\ \text{s.c. } X_0 = x_0, \quad X_{N+1} = x_T. \end{cases}$$

Pour montrer ce résultat, on remarque premièrement que la fonction objectif  $C$  est convexe, soumise à des contraintes linéaires, ce qui implique que les conditions nécessaires d'optimalité données par les conditions de Karush-Kuhn-Tucker sont des conditions nécessaires et suffisantes. En effet, le vecteur d'états intermédiaires  $\bar{X}$  est donc solution du problème d'optimisation considéré si et seulement si il existe  $\lambda = (\lambda_0, \lambda_{N+1}) \in \mathbb{R}^2$  tel que  $\bar{X}$  soit l'unique solution de

$$\nabla_X L(X, \lambda) = 0, \quad X_0 = x_0, \quad X_{N+1} = x_T, \quad (5.6)$$

où  $L: \mathbb{R}^{N+2} \times \mathbb{R}^2 \rightarrow \mathbb{R}$  est le lagrangien du problème, défini par

$$L(X, \lambda) = C(X) - \lambda_0(X_0 - x_0) - \lambda_{N+1}(X_{N+1} - x_T),$$

En développant (5.6), on obtient le système suivant

$$\begin{aligned} & \begin{cases} X_0 = x_0, \\ \nabla_b C_{i-1}(X_{i-1}, X_i) + \nabla_a C_i(X_i, X_{i+1}) = 0, \quad \forall i \in \{1, \dots, N\}, \\ X_{N+1} = x_T, \end{cases} \\ \iff & \begin{cases} X_0 = x_0, \\ -X_{i-1} + 2X_i - X_{i+1} = 0, \quad \forall i \in \{1, \dots, N\}, \\ X_{N+1} = x_T. \end{cases} \end{aligned} \quad (5.7)$$

Il nous faut donc montrer que  $\bar{X} = (\bar{X}_0, \dots, \bar{X}_{N+1})$  est la solution de (5.7). La première et la dernière équation du système (5.7) sont vérifiées par construction, et on a pour tout  $i \in \{1, \dots, N\}$

$$-\bar{X}_{i-1} + 2\bar{X}_i - \bar{X}_{i+1} = -(x_0 + (i-1)k) + 2(x_0 + ik) - (x_0 + k(i+1)) = 0.$$

Finalement, nous venons de montrer que la méthode Empirique peut être vue comme un cas particulier de la méthode Macro-Micro, dans laquelle les fonctions valeur sont modélisées via l'équation (5.5).



**Comparaisons.** Pour la méthode Macro-Micro, le problème (Macro) est résolu comme indiqué dans la section 5.2.2.1. Les  $N + 1$  problèmes de commande optimale du problème (Micro) sont résolus dans les deux cas par la méthode de tir simple, en utilisant les mêmes méthodes numériques que celles utilisées pour résoudre le problème (OCP). L'écart de ces deux méthodes à la solution de référence présentée en section 5.1 en termes de coût final et de déviation de la trajectoire d'état, pour les cinq scénarios déjà considérés dans la section précédente, est donné dans la table 5.4. De plus, les trajectoires d'état associées à ces cinq scénarios pour la méthode de tir simple, la méthode Macro-Micro ainsi que pour la méthode Empirique sont montrées dans la figure 5.9. Comme nous pouvons le constater, l'erreur de la méthode Macro-Micro par rapport à la méthode de tir simple est inférieur à 2g de carburant sur tout le WLTC, avec une moyenne de 0.88g, ce qui est largement acceptable du point de vue de l'application considérée. De plus, nous pouvons constater que la méthode Macro-Micro est en moyenne plus de neuf fois plus proche de la consommation optimale que la méthode Empirique, ce qui montre donc la pertinence du problème (Macro).

$x_0$	$x_T$	Erreur coût (g / %)		Déviation de l'état en $\ \cdot\ _{L^2}$ (· / %)	
		Macro-Micro	Empirique	Macro-Micro	Empirique
0.3	0.7	<b>0.42 / 0.045</b>	6.65/0.72	<b>0.42 / 0.05</b>	2.80 / 2.05
0.4	0.6	<b>0.34 / 0.039</b>	7.35/0.89	<b>0.22 / 0.01</b>	1.77 / 0.80
0.5	0.5	<b>1.00 / 0.129</b>	9.17/1.18	1.00 / 0.23	<b>0.61 / 0.08</b>
0.6	0.4	<b>1.71 / 0.244</b>	8.53/1.21	<b>0.75 / 0.12</b>	0.76 / 0.13
0.7	0.3	<b>0.91 / 0.143</b>	10.19/1.60	<b>0.58 / 0.07</b>	1.29 / 0.33
Moyenne		<b>0.88 / 0.120</b>	8.37/1.12	<b>0.59 / 0.1</b>	1.45 / 0.68

TABLE 5.4 – Comparaison entre la méthode Macro-Micro et la méthode Empirique pour les cinq expériences définies par les deux premières colonnes. La colonne « Erreur coût » correspond à la différence entre le coût de la méthode et celui de la méthode de tir simple. Similairement, les deux dernières colonnes comparent la déviation de la trajectoire d'état par rapport à celle donnée par la méthode de tir simple en norme  $L^2$ .

**Initialisation des fonctions de tir.** L'objectif est maintenant de montrer la pertinence de l'initialisation des fonctions de tir donnée par l'équation (5.3). Nous allons comparer cette initialisation à une initialisation fixée mais bien choisie ( $p = 500$ ), sur le premier intervalle de temps. Cette initialisation est bien choisie car pour tout  $x_0 \in [0.2, 0.8]$ , elle appartient à la zone d'évolution « intéressante » de la fonction de tir. La convergence moyenne du solveur est donnée dans la figure 5.10.

Comme nous pouvons le constater, l'initialisation donnée par l'équation (5.3) permet d'améliorer la convergence du solveur vers un zéro de la fonction de tir. De plus, en comparant l'itéré initial de la figure 5.10, il est intéressant de constater que  $S_1(p)$  avec  $p$  donnée par (5.3) est en moyenne plus de 10 fois plus petit que  $S_1(500)$ . Enfin, nous avons constaté que l'évaluation de la fonction de tir en l'initialisation proposée satisfait directement la tolérance industrielle requise dans près de 10% des cas.

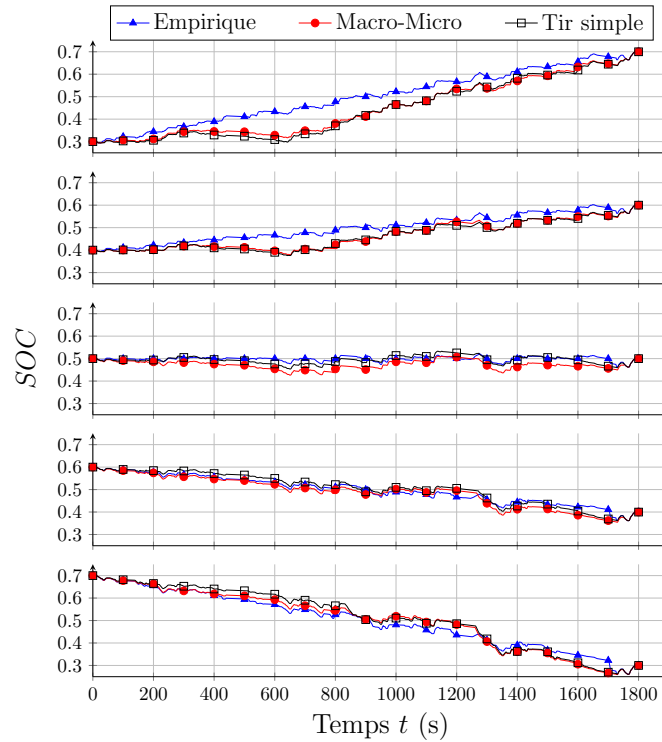


FIGURE 5.9 – Trajectoire d'état pour la méthode de tir simple, la méthode Empirique et la méthode Macro-Micro, pour les cinq scénarios considérés.

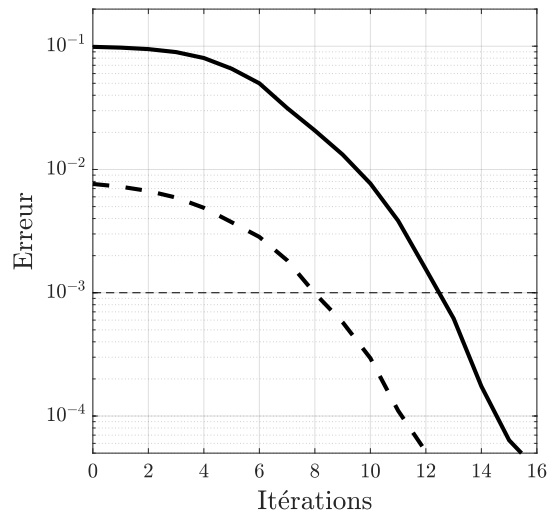


FIGURE 5.10 – Évolution de l'erreur  $|S_{0,a,b}(\cdot)|$  en fonction du nombre d'itérations. La courbe pleine correspond à une initialisation fixée  $p = 500$ , et la courbe discontinue à l'initialisation donnée par l'équation (5.3). L'optimisation est réussie si l'erreur est inférieure à  $10^{-3}$ , ce qui correspond à la ligne en pointillé noir. Ces données proviennent de la moyenne de 100 fonctions de tir  $S_{0,a,b}$  différentes, c'est-à-dire pour différentes conditions initiales  $a$  et finales  $b$  admissibles :  $(a, b) \in \dot{\Omega}_0$ .

### 5.2.4 Avantages

Nous venons d'appliquer la méthode Macro-Micro exposée dans le chapitre 3 au problème de répartition de couple dans les véhicules hybrides présenté dans le chapitre 1.

**Robustesse.** Comme nous l'avons constaté dans le chapitre 3 avec la figure 3.1, la méthode Macro-Micro est fortement liée à la méthode de tir multiple. Comme cette dernière est réputée pour être plus robuste que la méthode de tir simple, on peut s'attendre à ce que la méthode Macro-Micro hérite de cette meilleure robustesse. Par ailleurs, cette robustesse est encore améliorée dans l'application considérée si on utilise l'initialisation suggérée par l'équation (5.3).

**Temps et nombre de calcul.** La construction des bases de données  $\mathbb{D}_i$  et des modèles de fonctions valeur nécessaires à la méthode Macro-Micro est très coûteuse en temps de calcul, mais ce temps n'est pas comptabilisé ici, parce que cette tâche est réalisée une fois pour toutes et hors ligne comme on peut le voir dans la figure 5.5. Quant aux tâches à réaliser en ligne, le temps de calcul nécessaire à la résolution du problème (Macro) est négligeable ( $\approx 1$ s) et, comme discuté dans la section 3.2.5, il est suffisant de résoudre le premier des 18 problèmes (Micro) pour calculer la commande optimale courante. Il s'ensuit que la méthode Macro-Micro est  $N + 1 = 18$  fois plus rapide et nécessite  $(N + 1) = 18$  fois moins de calculs que la méthode de tir simple pour un horizon glissant de 18 intervalles de temps. Enfin, nous avons vu avec la figure 5.10 que l'initialisation proposée par l'équation (5.3) permet de réduire le nombre d'itérations du solveur, et donc d'accélérer encore la résolution.

**Optimalité.** Du fait de l'approximation des fonctions valeur  $V_i$  par des modèles  $C_i$ , la méthode Macro-Micro est sous-optimale. Cependant, comme nous l'avons vu à la section 3.2.4, il est possible de maîtriser l'erreur globale de la méthode Macro-Micro en contrôlant l'erreur d'approximation lors de la construction de ces modèles  $C_i$ . On peut constater dans la table 5.4 que l'écart moyen entre la méthode Macro-Micro et la solution de référence obtenue par tir simple est en moyenne de 0.88g de carburant dans l'application considérée, sur les 1800s du WLTC, ce qui est tout à fait acceptable d'un point de vue industriel.

## 5.3 Préconditionnement géométrique

Nous allons maintenant appliquer la méthode de preconditionnement géométrique proposée dans le chapitre 4 au problème de répartition de couple dans les véhicules hybrides. Plus précisément, l'idée est d'appliquer cette méthode sur les problèmes de commande optimale du problème (Micro). Ces problèmes sont de la même forme que le problème (OCP), c'est pourquoi nous allons utiliser sa formulation afin de simplifier les notations.

Nous allons dans un premier temps dans la section 5.3.1 appliquer directement cette méthode de preconditionnement, avec l'état initial fixé, et sans nous préoccuper du temps de calcul nécessaire à la construction de l'ensemble accessible augmenté. Nous

allons ensuite dans la section 5.3.2 nous rapprocher des hypothèses induites par la méthode Macro-Micro, c'est-à-dire que la condition initiale n'est pas considérée fixée. De plus, nous n'allons pas introduire de calculs coûteux supplémentaires pour générer des points de la frontière de l'ensemble accessible. Ces points seront uniquement ceux des bases de données  $\mathbb{D}_i$ .

### 5.3.1 Preuve de concept

Dans cette preuve de concept, nous allons directement appliquer la méthode de preconditionnement sur le problème (OCP) dans un cadre limité :  $t_0 = 0$ ,  $t_f = 100$  et  $x_0 = 0.5$ . Nous allons nous attacher à montrer l'efficacité du preconditionnement et de l'initialisation en termes de réduction du nombre d'itérations de la méthode de tir.

**Forme augmentée.** Ce problème de commande optimale sous forme augmentée s'écrit de la manière suivante

$$(AOC P_*) \quad \begin{cases} \min_{\hat{x}, u} x^0(t_f) \\ \text{s.c. } \dot{\hat{x}}(t) = \hat{f}(t, \hat{x}(t), u(t)), & t \in [t_0, t_f] \text{ p.p.}, \\ u(t) \in U(t), & t \in [t_0, t_f], \\ \hat{x}(t_0) = \hat{x}_0 := (0, x_0), \quad x(t_f) = x_T. \end{cases}$$

où le temps initial  $t_0 = 0$ , le temps final  $t_f = 100$  et la condition initiale  $x_0 = 0.5$  sont fixés. L'état augmenté  $\hat{x} = (x^0, x)$  est composé du coût  $x^0 \in AC([t_0, t_f], \mathbb{R})$  et de l'état  $x \in AC([t_0, t_f], \mathbb{R})$ . Dans le problème de répartition de couple considéré, le coût correspond à la trajectoire de consommation de carburant et l'état à la trajectoire d'état de charge de la batterie. Le système augmenté  $\hat{f}$  est donc constitué de

$$\hat{f}(t, \hat{x}, u) = (f^0(t, x, u), f(t, x, u)),$$

où  $f^0$  correspond à la consommation instantanée de carburant et  $f$  à la dynamique de l'état de charge de la batterie. Le pseudo-hamiltonien du système augmenté est défini par

$$\hat{h}(t, \hat{x}, \hat{p}, u) = (\hat{p} | \hat{f}(t, \hat{x}, u))$$

où  $\hat{p} = (p^0, p)$  est le co-état du système augmenté. L'application exponentielle de  $\vec{h}$  est définie directement à partir de la fonction `flot` (Équation (5.1)) par

$$\exp_{\vec{h}}(\hat{p}_0) = \left( (x^0(t_f), x(t_f)), (p^0, p(t_f)) \right),$$

avec  $\hat{p}_0 = (p^0, p_0)$ .

**Ensemble accessible et difféomorphisme.** La frontière de l'ensemble accessible du système augmenté est construite à partir de l'application exponentielle par

$$\partial \mathcal{A} = \left\{ \hat{x}_f \mid \exists \hat{p}_0 \neq 0, (\hat{x}_f, \hat{p}_f) = \exp_{\vec{h}}(\hat{p}_0) \right\}.$$

On construit numériquement l'ellipse  $\partial\mathcal{E}$  comme celle qui approche au mieux une discrétisation assez fine de  $\partial\mathcal{A}$ . On obtient ainsi par une méthode de moindres carrés les 5 paramètres  $c, a, b$ , et  $\theta$  correspondant respectivement au centre, aux axes et à l'inclinaison de l'ellipse. On construit ensuite le difféomorphisme linéaire  $\phi$  de  $\mathbb{R}^2$  satisfaisant l'équation (4.10) en utilisant le développement effectué dans l'exemple de la section 4.3.2. Ce difféomorphisme est donné par

$$\phi(\hat{x}) = r(-\beta_0)s(a, b)r(\theta)(\hat{x} - c), \quad \text{avec} \quad \beta_0 = \arctan\left(\frac{a \sin \theta}{b \cos \theta}\right). \quad (5.8)$$

Comme ce difféomorphisme est linéaire, on note  $\phi(\hat{x}) = A\hat{x} + b$ , avec

$$A = \begin{bmatrix} k & A_{x^0} \\ 0 & A_x \end{bmatrix} \quad \text{et} \quad b = \begin{bmatrix} b_{x^0} \\ b_x \end{bmatrix},$$

où  $k > 0$ ,  $A_{x^0}, A_x, b_{x^0}, b_x \in \mathbb{R}$ . Les ensembles  $\mathcal{A}$ ,  $\phi(\mathcal{A})$  et leurs approximations sont montrés dans la figure 5.11.

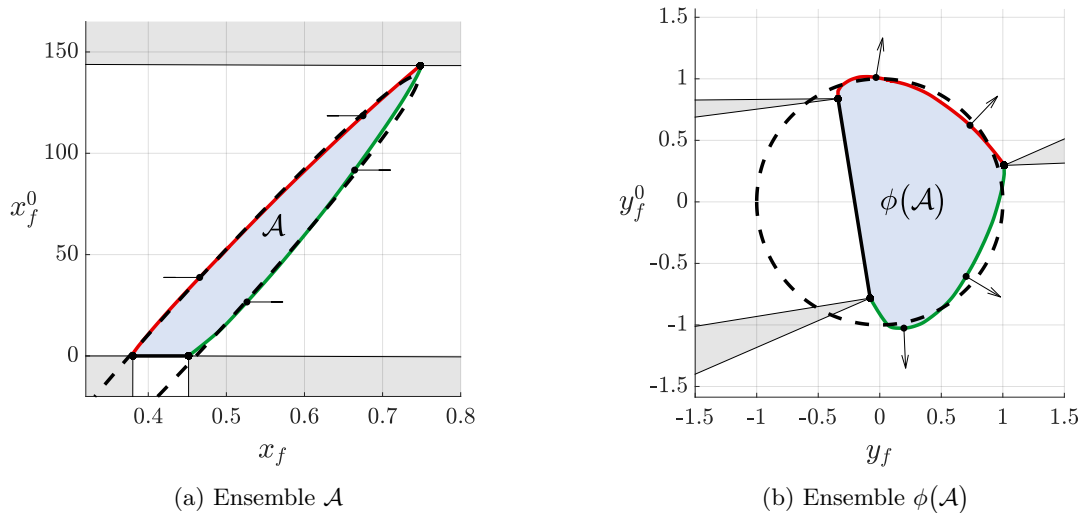


FIGURE 5.11 – Ensembles  $\mathcal{A}$ ,  $\phi(\mathcal{A})$  et leurs approximations pour  $t_0 = 0$ ,  $t_f = 100$  et  $x_0 = 0.5$ .

**Analyse de l'ensemble accessible.** Comme nous pouvons le constater, l'ensemble accessible du système augmenté  $\mathcal{A}$  est mal conditionné. En effet, on peut voir une différence d'échelle entre le coût  $x^0$  qui varie entre 0 et 160 et l'état  $x$  qui varie entre 0.45 et 0.75. De plus, on remarque que la somme des angles d'ouverture des cônes normaux associés aux coins représente quasiment tout le cercle (plus de 99,84%). Ce résultat est d'autant plus visible sur la figure 5.12.

À l'inverse, la figure 5.11b confirme que l'ensemble  $\phi(\mathcal{A})$  est mieux conditionné par construction. De plus, la somme des angles d'ouverture des cônes normaux associés aux coins de  $\partial(\phi(\mathcal{A}))$  est bien plus petite (15.15% du cercle seulement). Le pendant de la figure 5.12 dans les nouvelles coordonnées correspond à la figure 5.13.

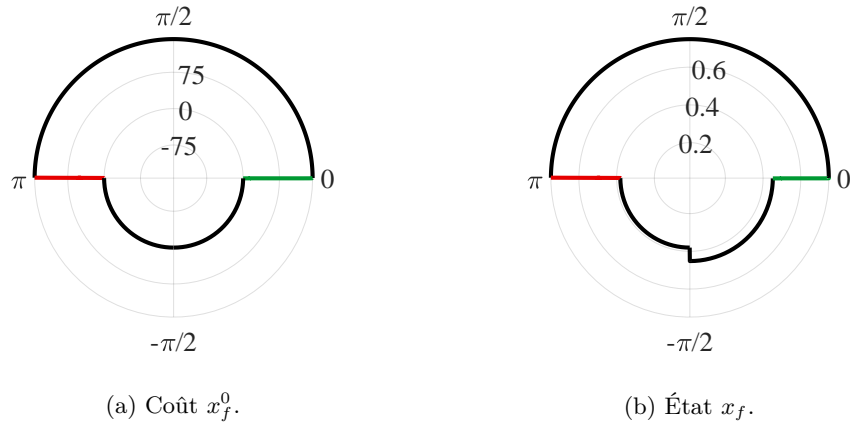


FIGURE 5.12 – Évolution de coût  $x_f^0$  (à gauche) et de l'état  $x_f$  (à droite), en fonction de l'angle du co-état augmenté final  $\hat{p}_f$  en représentation polaire, dans les coordonnées initiales.

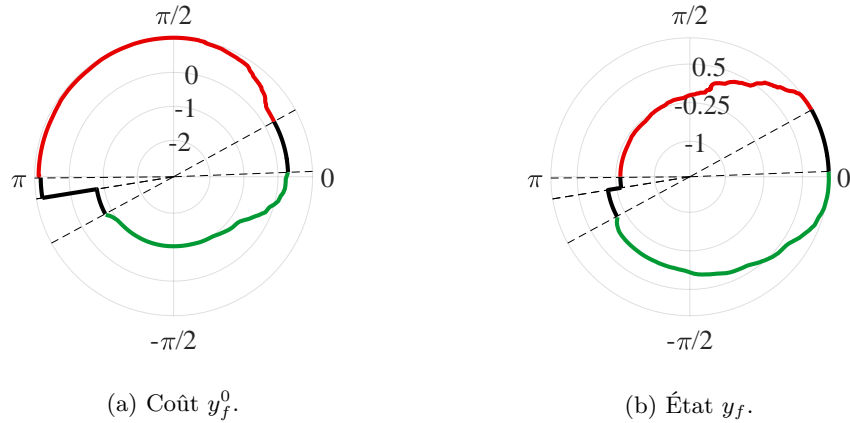


FIGURE 5.13 – Évolution de coût  $y_f^0$  (à gauche) et de l'état  $y_f$  (à droite), en fonction de l'angle du co-état augmenté final  $\hat{q}_f$  en représentation polaire, dans les nouvelles coordonnées.

**Correspondance des co-états.** Il nous faut maintenant construire la fonction  $\hat{\varphi}_0$  qui est l'approximation de la fonction  $\hat{p}_0(\cdot)$ , inverse de la fonction

$$\hat{p}_f(\hat{p}_0) = \pi_{\hat{p}}(\exp_{\vec{h}}(\hat{p}_0)).$$

Deux coupes de cette fonction sont montrées dans la figure 5.14, correspondant aux deux méthodes de normalisation proposées pour les fonctions de tir.

Comme nous pouvons le constater, la fonction  $\hat{p}_f(\cdot)$  est proche de la fonction identité sur les deux coupes qui nous intéressent. C'est pourquoi nous avons modélisé la fonction  $\hat{\varphi}_0$  par l'application identité.

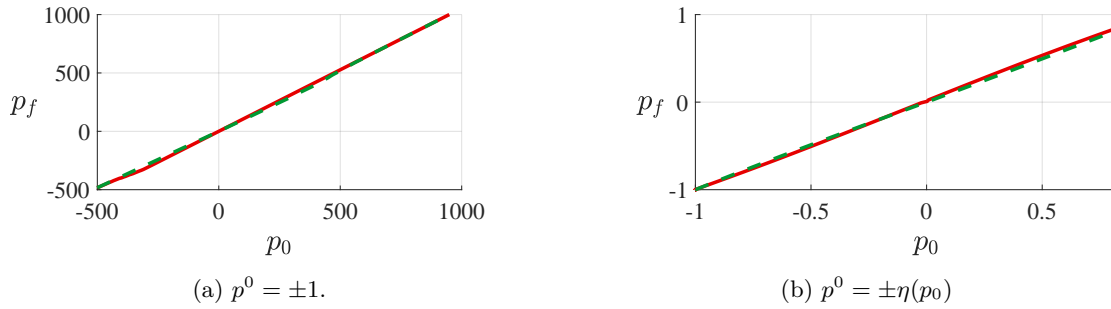


FIGURE 5.14 – Évolution du co-état final  $p_f$  en fonction du co-état initial  $p_0$ , avec  $t_0 = 0$ ,  $t_f = 100$  et  $x_0 = 0.5$ . La figure de gauche (respectivement droite) correspond à la première (respectivement seconde) méthode de normalisation de la fonction de tir.

**Fonctions de tir.** Il nous reste maintenant à construire les fonctions de tir, dans les anciennes coordonnées ( $S_1$  et  $S_2$ ) et dans les nouvelles coordonnées ( $T_1$  et  $T_2$ ). On rappelle que les deux fonctions de tir dans les coordonnées initiales  $S_1: \mathbb{R} \rightarrow \mathbb{R}$  et  $S_2: [-1, 1] \rightarrow \mathbb{R}$  sont construites à partir de la fonction  $S: \mathbb{R}^- \times \mathbb{R} \rightarrow \mathbb{R}$  définie par

$$S(\hat{p}_0) = \pi_x(\exp_{\vec{h}}(\hat{p}_0)) - x_T$$

par

$$S_1(p_0) = S(-1, p_0) \quad \text{et} \quad S_2(p_0) = S(\eta(p_0), p_0).$$

Dans les nouvelles coordonnées, la fonction  $T: \mathbb{R}^- \rightarrow \mathbb{R}$  est définie par

$$T(\hat{q}_f) = A_x \pi_x \left( (\exp_{\vec{h}} \circ \hat{\varphi}_0)(A^\top \hat{q}_f) \right) + b_x - y_T,$$

où  $y_T = A_x x_T + b_x$ . Comme  $\hat{\varphi}_0 = \text{Id}$ , on a donc

$$T(\hat{q}_0) = A_x \pi_x (\exp_{\vec{h}}(A^\top \hat{q}_0)) + b_x - y_T.$$

Les fonctions  $T_1: \mathbb{R} \rightarrow \mathbb{R}$  et  $T_2: [-1, 1] \rightarrow \mathbb{R}$  sont construites à partir de  $T$  de la manière suivante

$$T_1(q_0) = S(-1, q_0) \quad \text{et} \quad T_2(q_0) = S(\eta(q_0), q_0).$$

Ces quatre fonctions de tir sont représentées en figure 5.15.

**Résultats numériques.** La fonction qui nous intéresse est la fonction  $T_2$ . Nous avons montré dans le chapitre 4 que si  $\phi(\mathcal{A}) = \mathcal{B}$  et  $\hat{p}_f(\cdot) = \text{Id}(\cdot)$ , alors  $T_2$  est la fonction identité sur  $[-1, 1]$ . Ce cas idéal n'est toutefois qu'une approximation du problème industriel de répartition de couple. Néanmoins, si les approximations sont suffisamment précises, on peut s'attendre à ce que  $T_2$  soit proche de l'application Identité sur  $[-1, 1]$ . De plus, on remarque sur la figure 5.15 que les parties constantes (en noir) de cette fonction de tir sont bien moins étendues que celles de  $S_2$ . Cette caractéristique est due directement à la réduction de l'angle d'ouverture des cônes normaux associés aux coins de la frontière de l'ensemble  $\partial(\phi(\mathcal{A}))$ . Finalement, comme nous l'avons vu, une initialisation naturelle de cette fonction de tir est donnée par  $q_0 = y_T$ .

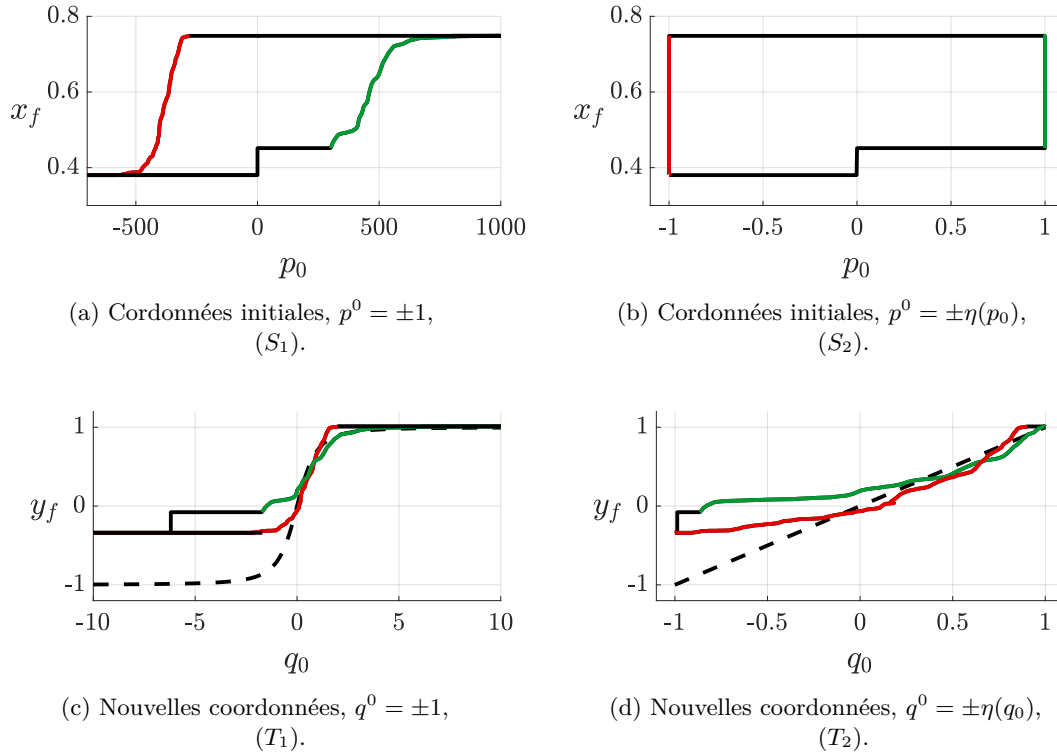


FIGURE 5.15 – Évolution de l'état final en fonction du co-état initial, dans les coordonnées d'origine (en haut) et dans les nouvelles coordonnées (en bas). Pour  $x_T$  (et donc  $y_T$ ) fixé, les fonctions de tir indiquées entre parenthèses sont obtenues par une translation des courbes vertes de  $x_T$  dans les coordonnées d'origine et de  $y_T$  dans les nouvelles coordonnées.

La recherche de zéro de  $T_2$  est effectuée de la même manière que celle de  $S_1$ . De plus, il est possible d'utiliser la bonne initialisation (dans les coordonnées d'origine) donnée par l'équation 5.3 dans les nouvelles coordonnées en prenant

$$q_0 = \frac{k_2}{\|k\|} \quad \text{avec} \quad k = \begin{bmatrix} k_1 \\ k_2 \end{bmatrix} = A^{-\top} \begin{bmatrix} -1 \\ -\nabla_a C_0(x_0, x_T) \end{bmatrix}. \quad (5.9)$$

La convergence moyenne du solveur pour ces deux fonctions est montrée dans la figure 5.16. Comme nous pouvons le voir, la convergence de la méthode de tir est plus rapide avec la fonction  $T_2$  (en rouge) qu'avec la fonction  $S_1$  (en noir). Le meilleur résultat est obtenu pour la fonction  $T_2$  et l'initialisation donnée par l'équation (5.9). De plus, en comparant la courbe discontinue noire et celle en trait plein rouge, on peut remarquer que le gain en efficacité est essentiellement dû au preconditionnement. Finalement, l'initialisation donnée par  $q_0 = y_T$  pour la fonction  $T_2$  (traits pointillés rouges) est assez bonne et permet une convergence rapide. De plus, cette initialisation fournit une alternative intéressante dans le cas où la fonction  $C_0$  n'est pas connue puisque l'utilisation de l'équation (5.9) nécessiterait de construire cette fonction, ce qui est très coûteux.



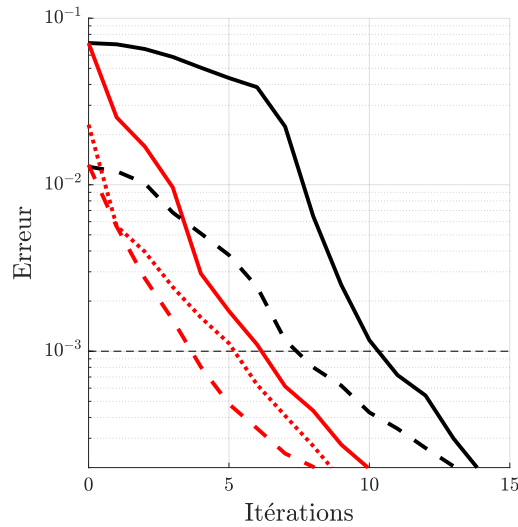


FIGURE 5.16 – Évolution de l’erreur (valeur absolue de la fonction de tir) en fonction du nombre d’itérations, avec  $t_0 = 0$ ,  $t_f = 100$  et  $x_0 = 0.5$ . Les courbes noires (resp. rouges) correspondent à la fonction de tir  $S_1$  (resp. à  $T_2$ ). Les courbes pleines correspondent à une initialisation fixée mais bien choisie ( $p_0 = 500$  pour  $S_1$  et  $q_0 = 0$  pour  $T_2$ ). Les courbes en trait discontinu correspondent à l’initialisation donnée par l’équation (5.3) pour  $S_1$  et l’équation (5.9) pour  $T_2$ . Finalement, la courbe en pointillés rouges correspond à l’initialisation donnée par  $q_0 = y_T$ . Ces données proviennent de la moyenne de 100 fonctions de tir différentes, c’est-à-dire pour différentes conditions finales  $x_T$  admissibles :  $(x_0, x_T) \in \Omega_0$ .

**Conclusion.** Nous venons d’appliquer la méthode de préconditionnement géométrique décrite dans le chapitre 4 au problème de répartition de couple dans les véhicules hybrides. Plus précisément, nous nous sommes intéressés au premier sous-intervalle de temps ( $t_0 = 0$  et  $t_f = 100$ ), avec l’état initial  $x_0 = 0.5$  fixé. Nous avons vu que cette méthode de préconditionnement géométrique accélère la résolution du problème considéré, pour tout état final  $x_T$  admissible. Cette méthode et les résultats associés sont généralisables sur tous les sous-intervalles de temps  $\Delta_i$  et pour toute condition initiale  $x_0$ .

### 5.3.2 Généralisation

La méthode de préconditionnement est généralisable pour tout état initial  $x_0$ , ainsi que pour tout intervalle de temps  $\Delta_i$ . Cependant, l’ensemble accessible dépendant de ces paramètres, dès qu’ils varient, il nous faut reconstruire l’ellipse  $\mathcal{E}$ , ainsi que le difféomorphisme  $\phi$  qui lui est associé, ce qui est assez coûteux. Dans cette section, tout en restant dans l’intervalle fixé  $\Delta_0$ , nous allons paramétrer la méthode de préconditionnement, et notamment  $\phi$ , par la condition initiale  $x_0$ . Nous allons voir que, dans le cas de notre application, la dépendance de l’ensemble accessible du système augmenté à  $x_0$  peut être traitée assez facilement.

Dans la section précédente, nous avons une connaissance de la totalité de la frontière de l'ensemble accessible (courbes verte et rouge dans la figure 5.11a), parce que nous avons construit des points spécifiquement à cet effet au moyen de l'application exponentielle pour de nombreux co-états augmentés initiaux. Afin de réduire les temps de calcul de l'approximation de  $\partial\mathcal{A}$  par une ellipse, on propose de se contenter de points déjà disponibles, à savoir ceux utilisés pour construire la base de données  $\mathbb{D}_0$ . Remarquons toutefois que dans cette base de données, seuls les points de la partie verte de la frontière sont représentés, ceux associés aux extrémales d'intérêt du problème de répartition de couple (qui minimisent le coût avec  $p^0 < 0$ ).

**Construction de l'ellipse.** Sur un trajet donné, l'énergie totale dépensée pour la traction du véhicule est la même, quelle que soit la stratégie de répartition de couple utilisé. On peut raisonnablement penser que la consommation de carburant dépend moins de l'état de charge initial  $x_0$  que de la variation d'état de charge  $\Delta x = x_f - x_0$ . Cette intuition est confirmée par la répartition des points de  $\mathbb{D}_0$  tracés en figure 5.17a en fonction de  $\Delta x$ .

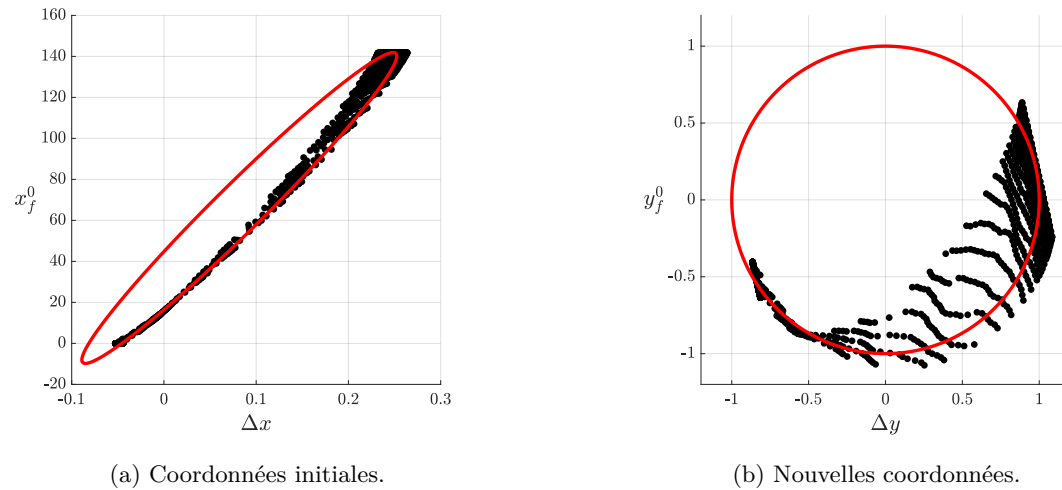


FIGURE 5.17 – À gauche, les points construits à partir de  $\mathbb{D}_0$  et l'ellipse ajustée. À droite, leur transformation par  $\phi_\Delta$ .

Notre idée est donc de faire passer une ellipse  $\mathcal{E}$  à travers ce nuage de points, puis d'en déduire le difféomorphisme  $\phi_\Delta(\Delta\hat{x})$  donné par l'équation (5.8). On obtient enfin pour tout  $\hat{x}_0$  le difféomorphisme  $\phi_{\hat{x}_0}$  de la manière suivante

$$\phi_{\hat{x}_0}(\hat{x}_f) = \phi_\Delta(\hat{x}_f - \hat{x}_0), \quad (5.10)$$

où  $\hat{x}_0 = (0, x_0)$ , avec  $x_0$  l'état initial. La transformation par  $\phi_\Delta$  des points de la base de donnée  $\mathbb{D}_0$  montrés dans la figure 5.17a sont représentés sur la figure 5.17b.

**Ensemble accessible et fonction de tir.** Afin de pouvoir évaluer les différences entre l'utilisation de  $\phi_{\hat{x}_0}$  et  $\phi$ , nous proposons de visualiser la transformation par  $\phi_{\hat{x}_0}$  de l'ensemble  $\mathcal{A}$  construit dans la section 5.3.1 pour lequel  $x_0 = 0.5$ . Les ensembles  $\mathcal{A}$  et  $\phi_{\hat{x}_0}(\mathcal{A})$  sont montrés dans la figure 5.18.

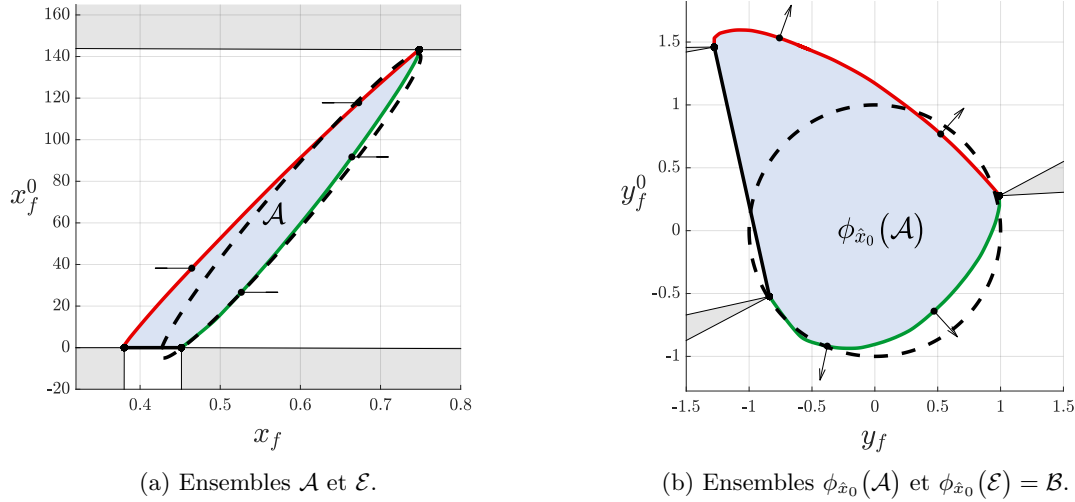


FIGURE 5.18 – Transformation des ensembles  $\mathcal{A}$  et  $\mathcal{E}$  par  $\phi_{\hat{x}_0}$ , avec  $t_0 = 0$ ,  $t_f = 100$  et  $x_0 = 0.5$ . Cette figure peut être comparée à la figure 5.11.

Les fonctions de tir  $T_1$  et  $T_2$  construites à partir de  $\phi_{\hat{x}_0}$  de la même manière que dans la section précédente sont affichés à travers la figure 5.19.

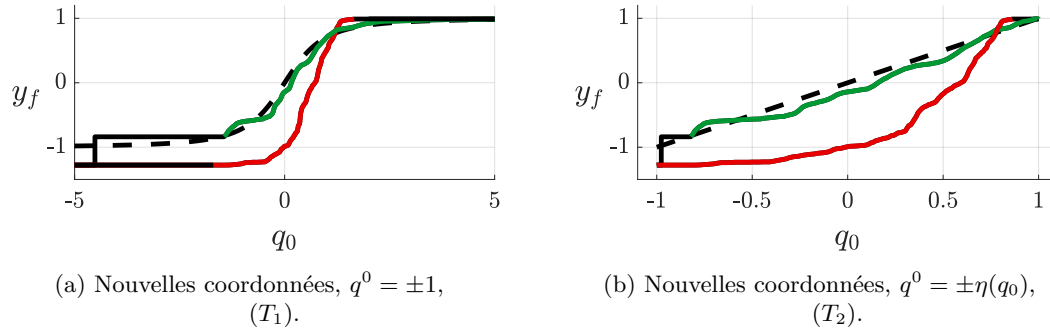


FIGURE 5.19 – Évolution de l'état final  $y_f$  en fonction du co-état initial  $q_0$  dans les nouvelles coordonnées. Les fonctions de tir indiquées entre parenthèses sont obtenues par une translation des courbes vertes de  $y_T$ . Cette figure peut être comparée à la figure 5.15.

Comme nous pouvons le voir, ces fonctions de tir sont plus proches des fonctions que l'on souhaiterait avoir (celles en trait discontinu noir) que les fonctions de tir construites dans la section précédente, c'est-à-dire celles de la figure 5.15. On considère que cette amélioration est due au fait que la base de donnée  $\mathbb{D}_0$  n'a été construite qu'à partir d'extrémales d'intérêt de notre problème, et non de la totalité de la frontière de l'ensemble admissible comme fait précédemment (où  $p^0$  pouvait être positif).

**Resultats.** Pour les mêmes expériences que celles effectuées dans la figure 5.16, nous avons étudié la convergence de la méthode de tir avec la fonction  $T_2$ , avec trois initialisations :  $q^0 = 0$ , celle donnée par l'équation (5.9), et également  $q_0 = y_T$ . Les résultats sont comparés à ceux obtenus pour la fonction  $S_1$  dans la figure 5.20.

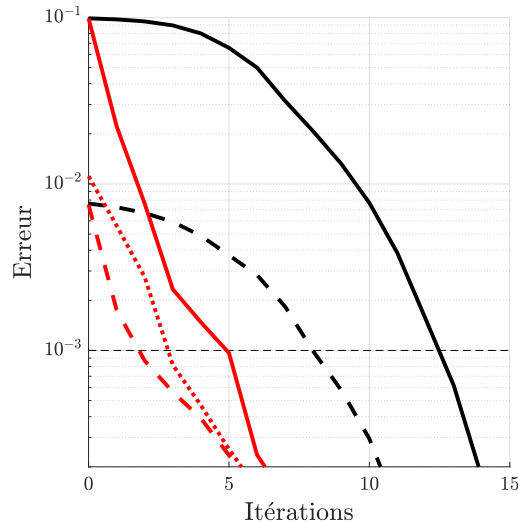


FIGURE 5.20 – Évolution de l'erreur (valeur absolue de la fonction de tir) en fonction du nombre d'itérations, avec  $t_0 = 0$ ,  $t_f = 100$ . Les courbes noires (resp. rouges) correspondent à la fonction de tir  $S_1$  (resp. à  $T_2$ ). Les courbes pleines correspondent à une initialisation fixée mais bien choisie ( $p_0 = 500$  pour  $S_1$  et  $q_0 = 0$  pour  $T_2$ ). Les courbes en trait discontinu correspondent à l'initialisation donnée par l'équation (5.3) pour  $S_1$  et l'équation (5.9) pour  $T_2$ . Finalement, la courbe en pointillé rouge correspond à l'initialisation donnée par  $q_0 = y_T$ . Ces données proviennent de la moyenne sur 100 fonctions de tir différentes, c'est-à-dire pour différentes conditions initiales  $x_0$  et finales  $x_T$  admissibles :  $(x_0, x_T) \in \mathring{\Omega}_0$ .

Comme nous pouvons le constater, la méthode de preconditionnement géométrique permet au solveur de converger plus rapidement que la méthode de tir classique. De plus, en comparant cette figure à la figure 5.16, on remarque une amélioration de la convergence dans cette section. Comme remarqué précédemment, cela est très probablement dû au fait que nous avons utilisé uniquement des points associés aux extrémales d'intérêt de notre problème (dans  $\mathbb{D}_0$ ) pour créer l'ellipse  $\mathcal{E}$ .

**La méthode de preconditionnement combinée à l'initialisation donnée par l'équation (5.9) permet au solveur de converger en seulement deux itérations en moyenne sur l'application considérée.**

C'est le principal résultat numérique associée à la méthode de preconditionnement.

## Conclusion

Dans ce chapitre, nous venons d'appliquer la méthode de tir simple ainsi que la méthode Macro-Micro au problème de répartition de couple dans les véhicules hybrides décrit dans le chapitre 1. Ces deux méthodes ont été comparées en termes d'optimalité, de temps de calcul, de nombre de calculs et de robustesse. De plus, les problèmes de commande optimale du problème (Micro) ont été résolus avec une liste d'états intermédiaires générée d'une part par la méthode Empirique et d'autre part par la résolution du problème (Macro), ce qui a permis de vérifier l'intérêt de la méthode Macro-Micro.

Nous avons ensuite appliqué la méthode de préconditionnement géométrique, décrite dans le chapitre 4, au premier problème de commande optimale du problème (Micro), c'est-à-dire au problème (OCP) avec  $t_0 = 0$ ,  $t_f = 100$  et  $x_0$  connu. Nous nous sommes ensuite placés, dans la section 5.3.2, sous les mêmes hypothèses que celles des problèmes de commande optimale du problème (Micro), à savoir que l'état initial n'est pas connu avant d'avoir effectué la résolution du problème (Macro). Nous avons donc construit une méthode de préconditionnement paramétrée par cette condition initiale. De plus, seules des données préalablement calculées pour construire les approximations des fonctions valeur ont été utilisées pour construire une ellipse unique  $\mathcal{E}$ .

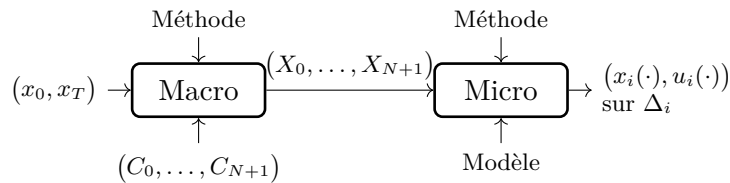
La méthode de préconditionnement géométrique se combine à la méthode Macro-Micro sans introduire de coût numérique supplémentaire. Il nous faut simplement, sur chaque intervalle de temps  $\Delta_i$ , approcher par une ellipse un ensemble de points déjà calculés pour générer la base de données  $\mathbb{D}_i$ . Cette combinaison mène à une méthode performante : en plus d'allier les avantages de la méthode Macro-Micro décrits dans la section 5.2.4, la méthode de préconditionnement géométrique permet au solveur de converger en seulement deux itérations en moyenne. On dispose ainsi d'une méthode de résolution des problèmes de commande optimale du problème (Micro) plus rapide et moins coûteuse qu'avec la méthode de tir simple classique, et compatible avec une solution embarquée.

## Perspectives

---

<b>6.1</b>	<b>Multi-cycle</b> .....	<b>124</b>
6.1.1	Hypothèses .....	124
6.1.2	Résolution .....	124
<b>6.2</b>	<b>Connaissance partielle</b> .....	<b>125</b>
6.2.1	Hypothèses .....	126
6.2.2	Résolution .....	127
<b>6.3</b>	<b>Macro-Micro-ECMS</b> .....	<b>127</b>
6.3.1	Méthode ECMS .....	128
6.3.2	Variante proposée .....	128

La méthode (Macro)-(Micro) introduite dans le chapitre 3 fait partie d'une famille de méthodes à deux niveaux schématisée ci-dessous



Cette famille de méthodes à deux niveaux est paramétrée par les méthodes utilisées pour résoudre le problème (**Macro**) et le problème (**Micro**), par la manière dont on obtient les approximations des fonctions valeur  $C_i$  ainsi que par le modèle utilisé pour le problème (**Micro**).

Ce dernier chapitre est consacré aux perspectives d'amélioration possibles pour cette famille de méthode à deux niveaux et à la possible adaptation de la méthode de préconditionnement géométrique au problème de répartition de couple. Plus précisément, nous reviendrons sur des hypothèses qui ne sont pas réalistes dans le cadre d'une application embarquée, principalement celle qui suppose la connaissance totale et parfaite du cycle. Cette hypothèse sera remise en cause dans ce chapitre et en fonction des nouvelles hypothèses introduites, nous allons discuter des possibles solutions qui peuvent être envisagées.

Nous allons considérer dans un premier temps dans la section 6.1 que le cycle n'est plus uniquement le WLTC, et dans un second temps dans la section 6.2 que le cycle n'est pas connu entièrement à partir du temps initial. Finalement, nous proposons dans la section 6.3 une nouvelle méthode qui combine la méthode Macro-Micro et la méthode ECMS.

## 6.1 Multi-cycle

Les hypothèses considérées précédemment pour le problème de répartition de couple ne sont pas réalistes pour fournir une solution embarquée dans un véhicule. Plus précisément, nous nous sommes limités à un cycle unique, à savoir le WLTC, que nous considérons connu et fixé avant même de proposer une méthode de résolution.

### 6.1.1 Hypothèses

**Connaissance.** Considérons maintenant que le cycle n'est plus unique mais que nous le connaissons parfaitement et entièrement au démarrage du véhicule. Plus précisément, nous connaissons donc la vitesse  $v(t)$  du véhicule ainsi que la pente de la route  $p(t)$  pour tout temps  $t \in [0, t_f]$ , où le temps final  $t_f$  dépend également du cycle.

Comme nous l'avons fait pour le WLTC dans la section 1.1.1, il est possible de déduire la vitesse de rotation des roues  $N_w(t)$  ainsi que le couple imposé aux roues  $T_{qW}(t)$ , pour tout  $t \in [t_0, t_f]$ . Ces deux fonctions sont donc considérées connues au démarrage du véhicule.

**Fonctions.** Supposons également que nous disposons de deux fonctions, la fonction  $C$  et la fonction  $E$ , qui, à partir d'un sous-cycle de longueur  $l$ , nous fournissent respectivement

- une approximation  $C_i$  de la fonction valeur,
- les paramètres  $P_i = (c, a, b, \theta)$  de l'ellipse  $\mathcal{E}_i$  (c.f. Section 5.3.2)

associée à ce sous-intervalle de temps.

**Remarque.** Une méthode possible pour construire ces deux fonctions est d'utiliser des réseaux de neurones convolutifs, ou *Convolutional Neural Network* (CNN). Ces réseaux de neurones utilisent des produits de convolution à la place du produit classique et peuvent prendre en entrée des séries temporelles de taille fixe. Cependant, afin que ces deux CNN  $C$  et  $E$  soient performants, c'est-à-dire que leur erreur de prédiction soit faible, il faut les entraîner sur une grande base de données. On pourrait imaginer qu'à partir d'une base de données constituée de sous-cycle de longueur  $l$ , il est possible de créer une base de données  $\mathbb{D}$  comme étant l'union des bases de données  $\mathbb{D}_i$  créées avec la méthode 2 décrite dans la section 5.2.1.1 associées à chacun des sous-cycles. La création de cette grande base de données  $\mathbb{D}$  est sûrement une tâche très longue. Pour cette raison, mais également parce que nous n'avons pas eu accès à une base de donnée conséquente de cycles admissibles pour notre véhicule, cette perspective n'a pas pu être testée dans cette thèse.

### 6.1.2 Résolution

La connaissance de ces deux fonctions  $C$  et  $E$  rend possible l'application de la méthode Macro-Micro ainsi que de la méthode de préconditionnement géométrique à bord du véhicule. Un schéma général est donné dans la figure 6.1.

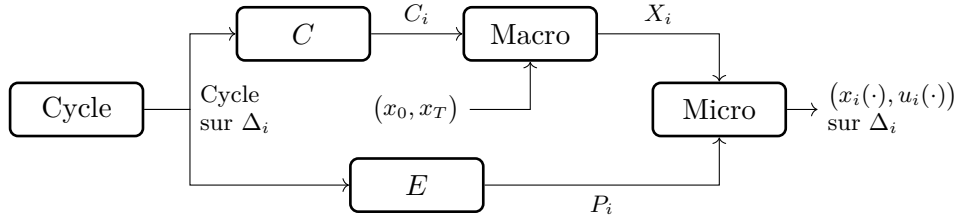


FIGURE 6.1 – Schéma des méthodes proposées.

On propose de décrire ce schéma étape par étape :

- le bloc « Cycle » fournit les informations provenant du cycle ainsi que le temps final  $t_f$ . Ces informations sont résumées aux deux séries temporelles considérées, à savoir la vitesse de rotation des roues  $N_w$  ainsi que le couple aux roues  $T_{qW}$ . De plus, ce bloc fournit ces informations sous forme de sous-cycle de longueurs  $l$ , définis sur les intervalles de temps  $\Delta_i$  pour tout  $i \in \mathbb{N}_N$ .
- le bloc «  $C$  » prend les couples de séries temporelles définies sur  $\Delta_i$  et fournit les fonctions  $C_i$  qui correspondent à des approximations des fonctions valeur associées aux sous-intervalles de temps  $\Delta_i$ , pour tout  $i \in \mathbb{N}_N$ .
- le bloc « Macro » résout le problème (Macro) en utilisant les méthodes décrites dans la section 5.2.2.1. Ce bloc prend donc en entrée les fonctions  $C_i$ , l'état initial  $x_0$  connu au démarrage de la voiture ainsi que l'état final  $x_T$  que l'on souhaite atteindre à la fin du cycle. Il fournit le vecteur d'états intermédiaires  $X = (X_0, \dots, X_{N+1})$  solution numérique du problème (Macro).
- le bloc «  $E$  » évalue la fonction  $E$  sur chaque couple de séries temporelles définies sur l'intervalle de temps  $\Delta_i$ . Il fournit donc pour chaque intervalle de temps  $\Delta_i$  les paramètres  $P_i$  de l'ellipse  $\mathcal{E}_i$ .
- le bloc « Micro » résout les problèmes de commande optimale du problème (Micro) en cherchant un zéro de la fonction  $T_2$  décrite dans la section 5.3.2 sur chaque intervalle de temps  $\Delta_i$ , avec  $X_i$  comme état initial et  $X_{i+1}$  comme état final. Ce bloc fournit donc les trajectoires d'état et de commande, solution des problèmes de commande optimale du problème (Micro).

Finalement, si nous connaissons les fonctions  $C$  et  $E$ , il est possible d'appliquer les deux méthodes proposées, sous les hypothèses considérées. De plus, si l'erreur de prédiction de ces deux fonctions est assez petite, il est possible d'obtenir des résultats semblables à ceux du chapitre 5.

## 6.2 Connaissance partielle

Supposer la connaissance parfaite et totale du cycle qu'une voiture va effectuer à son démarrage reste une hypothèse non réaliste. Des méthodes pour y parvenir ont toutefois été développées pour en avoir une connaissance plus ou moins précise :

- Un modèle statistique basé sur les chaînes de Markov non homogènes (dépendance en position) développé dans [37] permet de prédire les trajectoires de vitesse et d'accélération du véhicule grâce aux données réelles enregistrées.



- Une approche par clusters est développée dans [44]. On construit un modèle probabiliste du trafic et on concentre l'information de vitesse/accélération dans l'appartenance à un cluster. En condition réelle, on identifie le cluster courant à partir des informations de trafic et on en déduit des densités de trajectoires de vitesse/accélération. Il existe également des approches de clusters basées sur des règles expertes [28].
- Une autre approche est basée sur le partage en temps réel des données [12, 76] (vitesse des véhicules voisins, historique, capteurs, feux de circulations ...). L'idée est de récupérer les données des véhicules en temps réel, de calculer une prévision dans un serveur et de fournir l'information aux véhicules. De telles approches ont été développées [32], en utilisant des méthodes à base d'intelligence artificielle [31].

Nous allons maintenant relâcher cette hypothèse de connaissance partielle du cycle et proposer une idée permettant d'adapter les deux méthodes proposées dans cette thèse, à savoir la méthode Macro-Micro et le préconditionnement géométrique, à de nouvelles hypothèses.

### 6.2.1 Hypothèses

**Connaissance.** Supposons maintenant qu'au démarrage de la voiture, nous connaissons un ensemble d'informations, noté  $\theta$ . Ces informations peuvent être par exemple :

- la route empruntée par le conducteur, grâce à un GPS,
- les vitesses moyennes sur chacune des portions des routes,
- l'état de la circulation actuelle (la vitesse des autres véhicules, densité, état des feux de signalisation, accidents ...),
- des informations sur le conducteur actuel, comme par exemple le type de conduite (souple, sportif ...)

Dans un monde de plus en plus connecté, cette hypothèse nous semble assez réaliste, ou du moins pourrait l'être dans un futur assez proche. De plus, nous supposons qu'il est possible d'estimer le temps final  $t_f$  tout comme le font déjà la plupart des GPS actuels.

**Fonctions.** En plus de connaître les deux fonctions  $C$  et  $E$  introduites dans la section précédente, on suppose également, qu'à partir des informations  $\theta$ , il est possible de construire une fonction, noté  $\tau$ , qui nous fournit une prévision précise du cycle sur le premier sous-intervalle de temps  $\Delta_0 = [0, l]$ . En utilisant des informations de notre véhicule, on considère donc que cette fonction nous fournit le couple  $T_{qW}(t)$  ainsi que la vitesse de rotation  $N_W(t)$  des roues, pour tout  $t \in [0, l]$ .

On suppose également qu'à partir des informations  $\theta$ , il est possible de construire une fonction, noté  $\tilde{C}$ , qui nous fournit un ensemble de fonction  $\{C_1, \dots, C_N\}$  ainsi qu'une suite de temps intermédiaires  $l = t_1 < t_2 < \dots < t_N + 1 = t_f$  tels que  $C_i$  correspond à une estimation des fonctions valeur associées aux intervalles  $\Delta_i = [t_i, t_{i+1}]$ .

**Remarque.** On remarque que les temps intermédiaires sont des variables. Dans les méthodes présentées précédemment, nous avons fixé les temps intermédiaires au préalable avec une discrétisation uniforme de  $\Delta = [t_0, t_f]$ . Ce choix arbitraire n'est pas une obligation dans la Méthode Macro-Micro.

**Remarque.** Imaginons que nous disposions d'une classification de  $k$  labels des routes et de leur état (« autoroute », « nationale bouchée », ...) et d'une méthode pour prédire le label de la route emprunté à partir de  $\theta$  sur chaque pas de temps. Dans ce cas, la fonction  $\hat{C}$  pourrait fournir les temps  $t_i$  associés au changement de label, ainsi que les fonctions  $C_i$  qui correspondent à la moyenne de consommation optimale du véhicule sur le type de route  $k_i$  et sur l'intervalle de temps  $\Delta_i = [t_i, t_{i+1}]$ .

### 6.2.2 Résolution

Sous ces hypothèses, il est possible d'appliquer la méthode Macro-Micro ainsi que la méthode de préconditionnement géométrique de manière embarquée, comme nous pouvons le constater via le schéma de la figure 6.2.

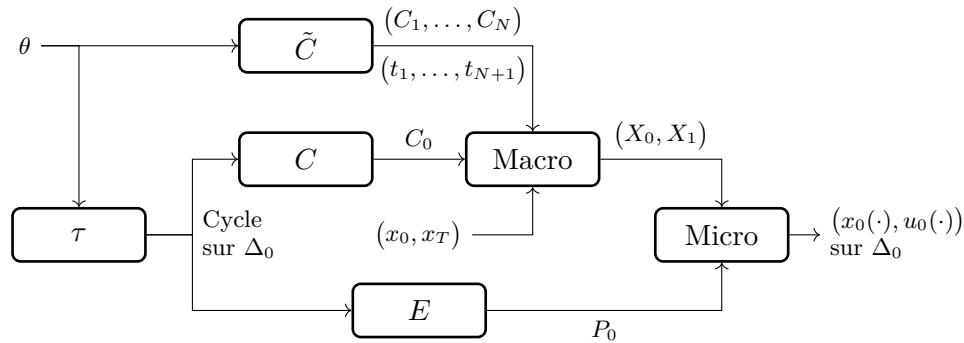


FIGURE 6.2 – Schéma des méthodes proposées.

Dans ce schéma, les blocs «  $C$  », «  $E$  », « Macro » et « Micro » correspondent aux blocs introduits dans la section précédente. De plus,

- le bloc «  $\tilde{C}$  » prend en entrée les informations  $\theta$  et fournit un ensemble de temps intermédiaires  $l = t_1 < \dots < t_{N+1} = t_f$  associés à un ensemble de fonctions  $C_1, \dots, C_N$ , où  $C_i$  correspond à une approximation de la fonction valeur associée à l'intervalle de temps  $\Delta_i = [t_i, t_{i+1}]$ ,
- le bloc «  $\tau$  » fournit, à partir des informations  $\theta$ , une prévision précise du cycle sur l'intervalle de temps  $\Delta_0 = [0, l]$  à partir des informations  $\theta$ .

On remarque que la méthode proposée fournit uniquement la trajectoire d'état  $x_0$  et de commande  $u_0$  à appliquer sur le premier intervalle de temps  $\Delta_0 = [0, l]$ . En effet, dans le cadre d'une solution embarquée, il est plus pertinent de construire une méthode qui sera relancée plusieurs fois durant la progression du cycle, afin de fournir une solution plus précise dans le sens où les informations  $\theta$  changeront sûrement au fil du temps.

## 6.3 Macro-Micro-ECMS

Comme nous l'avons décrit rapidement dans la section 1.4, la méthode ECMS introduite dans [53] est une méthode développée pour le problème de répartition de couple dans les véhicules hybride, et qui est largement utilisé pour ce problème. Une perspective intéressante serait de la comparer à la méthode Macro-Micro en termes d'optimalité, de

robustesse, de temps et de coût de calculs. Il est par ailleurs possible de combiner ces deux méthodes en une nouvelle que l'on nomme Macro-Micro-ECMS et que nous allons esquisser dans cette section. Pour cela, nous allons d'abord décrire plus précisément ce qu'est la méthode ECMS et établir son lien avec les méthodes indirectes. Nous décrirons ensuite le principe de la nouvelle méthode Macro-Micro-ECMS.

### 6.3.1 Méthode ECMS

L'idée centrale de la méthode ECMS est de construire un coût qui correspond à la somme de la consommation en carburant et de la déviation de l'état de charge, pondérée par un facteur équivalent noté  $\lambda$ . On s'intéresse donc à la résolution du problème

$$(ECMS) \quad \begin{cases} \min_{x,u} \int_{t_0}^{t_f} f^0(t, x(t), u(t)) + \lambda f(t, x(t), u(t)) dt, \\ \text{s.c.} \quad \dot{x}(t) = f(t, x(t), u(t)), \quad t \in [t_0, t_f] \text{ p.p.}, \\ \quad \quad u(t) \in U(t), \quad t \in [t_0, t_f], \\ \quad \quad x(t_0) = x_0. \end{cases}$$

En considérant  $\lambda \in \mathbb{R}$  fixé, il s'agit de trouver le contrôle admissible  $u(t) \in U(t)$  qui minimise l'intégrande du coût à chaque pas de temps. Cette méthode est donc très peu coûteuse et peut facilement être utilisée comme une méthode embarquée. On remarque cependant que l'état final n'est pas fixé : la difficulté est alors de trouver  $\lambda \in \mathbb{R}$  tel que si  $(x, u)$  est la trajectoire donnée par la méthode ECMS alors  $x(t_f) = x_T$ . La recherche de ce facteur équivalent  $\lambda$  peut se faire par la « méthode de tir » en cherchant un zéro de la fonction

$$S(\lambda) = x_f(\lambda) - x_T$$

où la fonction  $x_f(\cdot)$  correspond à l'état au temps  $t_f$  solution du problème suivant

$$\begin{cases} \dot{x}(t) = f(t, x(t), u(t)), & t \in [t_0, t_f] \text{ p.p.}, \\ u(t) \in \arg \max_{w \in U(t)} \{ f^0(t, x(t), w) + \lambda f(t, x(t), w) \} & t \in [t_0, t_f] \end{cases}$$

Trouver un zéro de  $S(\cdot)$  est de même complexité que trouver un zéro de la fonction de tir dans la méthode de tir simple. Dans la méthode ECMS, on réduit cette complexité en ne cherchant pas un zéro de la fonction  $S(\cdot)$  mais on se contentant d'une valeur de  $\lambda$  qui conduit à un état final  $x_f$  proche de la cible voulue. Les nombreuses variantes proposées pour cette méthode portent sur la détermination de  $\lambda$  et de sa possible évolution au cours du temps.

### 6.3.2 Variante proposée

Un lien assez fort [62, 75] a été établi entre la méthode ECMS et les méthodes indirectes. En effet, sous les deux hypothèses suivantes

- la fonction  $f^0$  ne dépend pas de  $x$  (ce qui est le cas dans notre formulation),
- la fonction  $f$  ne dépend pas de  $x$  (ce qui n'est pas le cas dans notre formulation),

la méthode ECMS correspond directement à l'application du principe du maximum, où l'on remarque que le co-état  $p(\cdot)$  est constant, noté  $p(\cdot) = p$ . La relation entre le facteur équivalent et le co-état est donné par  $\lambda = -p$ . Notre idée consiste simplement à appliquer la méthode Macro-Micro sur le problème (OCP) en remplaçant la méthode de tir simple utilisée pour résoudre les problèmes de commande optimale du problème (Micro) par la méthode ECMS. La valeur du facteur équivalent associé au  $i^{\text{ème}}$  problème de commande optimale du problème (Micro) serait alors donnée par

$$\lambda_i = -\nabla_a C_i(X_i, X_{i+1}),$$

où  $X = (X_0, \dots, X_{N+1})$  est une solution du problème (Macro). Un schéma illustratif de la méthode Macro-Micro-ECMS est proposé dans la figure 6.3.

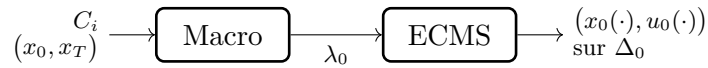


FIGURE 6.3 – Schéma de la méthode Macro-Micro-ECMS.

Comme nous pouvons le constater, cette méthode fournit les trajectoires d'état et de contrôle associées au premier intervalle de temps. Elle doit donc être relancée au fil de l'eau afin de mettre à jour le facteur équivalent.

Cette méthode, présentée dans le contexte du chapitre 5, peut être facilement adaptée aux méthodes introduites dans les sections 6.1 et 6.2.

## Conclusion

Nous venons de proposer trois méthodes de résolution différentes pour le problème de répartition de couple, en fonction des différentes hypothèses relatives à la connaissance du cycle. Comme nous l'avons remarqué en introduction, ces trois méthodes font, de notre point de vue, partie des méthodes à deux niveaux introduites dans le chapitre 3. En plus de fournir des méthodes rapides, moins coûteuses que les méthodes indirectes classiques et potentiellement quasiment optimales, les méthodes à deux niveaux peuvent s'adapter à différents niveaux d'hypothèses grâce à l'utilisation des approximations des fonctions valeur  $C_i$  qui permettent de tirer parti de toute l'information disponible afin de fournir des prévisions les plus précises possibles. Nous avons également proposé des solutions permettant d'adapter la méthode de préconditionnement géométrique à ces hypothèses dans le cas où les problèmes de commande optimale du problème (Macro) sont résolus par des méthodes indirectes.

Nous avons finalement proposé la méthode Macro-Micro-ECMS, qui est une combinaison de la méthode Macro-Micro développée dans cette thèse, et la méthode ECMS bien connue dans la littérature. En plus de tirer parti des avantages de ces deux méthodes en termes d'embarquabilité, la méthode Macro-Micro-ECMS s'appuie sur leur complémentarité en utilisant la capacité de la méthode Macro-Micro à fournir une valeur initiale pour le facteur équivalent  $\lambda$ , habituellement difficile à obtenir.



## Conclusion

---

**Bilan des contributions.** Le fil conducteur de cette étude, motivée par le problème de répartition de couple et du rapport de transmission de la boîte à vitesses dans les véhicules hybrides électriques, est le développement de nouvelles méthodes de contrôle optimal indirectes permettant de se rapprocher des critères d'embarquabilité (rapides, robustes et légères).

La première contribution de cette thèse correspond à la proposition de la méthode Macro-Micro [21] développée dans le chapitre 3. Cette méthode, basée sur une décomposition à deux niveaux du problème de commande optimale, fait intervenir explicitement des fonctions valeur de Bellman à des temps intermédiaires fixés. L'idée principale de cette méthode est d'utiliser des approximations de ces fonctions valeur permettant une résolution plus rapide, robuste et légère que les méthodes indirectes classiques. En effet, la méthode Macro-Micro revient à résoudre successivement un problème d'optimisation en dimension finie, le problème (Macro), et un ensemble de problèmes de commande optimale indépendants, le problème (Micro), définis sur des intervalles de temps plus courts. En outre, le théorème 3.1.5 nous permet d'obtenir une initialisation naturelle des fonctions de tir associées aux problèmes de commande optimale du problème (Micro), rendant leurs résolutions encore plus rapides et robustes.

La méthode de préconditionnement géométrique [23] de la fonction de tir proposée dans le chapitre 4 constitue la deuxième principale contribution de cette étude. L'idée principale de cette méthode est d'approcher la frontière de l'ensemble accessible du système augmenté par une ellipse et d'en déduire le difféomorphisme linéaire qui transforme cette ellipse en le cercle unité. Ce difféomorphisme est utilisé pour construire une fonction de tir qui, en utilisant l'interprétation géométrique du co-état et la transformée de Mathieu, devrait être assez proche d'une fonction affine, ce qui permet à un solveur de type Newton de trouver un zéro de cette fonction de tir en un nombre faible d'itérations. Les méthodes indirectes de résolution des problèmes (Micro) utilisant ce préconditionnement sont donc plus rapides et robustes, tout en restant optimales.

La troisième contribution principale de ce travail, proposée au chapitre 5, correspond à l'application de ces deux méthodes (ainsi que de la méthode de tir simple) au problème de répartition de couple décrit dans le chapitre 1. Les résultats obtenus sont très satisfaisants : d'une part, la méthode Macro-Micro conduit à une solution proche de la solution optimale, et d'autre part, la méthode de préconditionnement géométrique permet une convergence en seulement deux itérations en moyenne. Bien que ces deux méthodes aient été présentées et mises en œuvre de manière indépendante, elles peuvent être combinées, ce qui permet d'obtenir une méthode de résolution combinant les avantages de la méthode Macro-Micro décrit dans la section 5.2.4 et ceux du préconditionnement géométrique qui permet une convergence du solveur en seulement deux itérations en moyenne.

Finalement, d'autres contributions plus modestes sont proposées dans cette étude, à savoir principalement le nouveau point de vue sur la méthode de tir multiple interprétée comme étant l'application de la méthode de tir simple sur le problème parallélisé, proposé dans la section 2.3, ou bien la méthode efficace permettant de générer une base de données d'évaluation de la fonction valeur [22] décrite dans la section 5.2.1.1.

**Perspectives.** En plus des perspectives proposées dans le chapitre 6, centrées sur le problème de répartition de couple, de nombreux autres développements possibles méritent d'être mentionnés.

De manière générale, nous avons choisi un cadre de développement simple, propice aux méthodes développées et dans lequel le problème de répartition de couple s'inscrit. Nous avons donc considéré des hypothèses qui peuvent être vues comme assez restrictives tout au long du document. Une amélioration possible serait de développer des méthodes similaires dans un cadre plus général, où ces hypothèses seraient moins contraignantes. Par exemple, lors du développement de la méthode Macro-Micro, nous avons considéré que les fonctions valeurs étaient différentiables à la solution. Il existe cependant peu de résultat théorique sur l'existence de fonction valeurs avec une telle régularité, là où il existe des résultats pour des fonctions valeurs Lipschitz continues [71] ou  $\mathcal{C}^1$  par morceaux. Il serait donc intéressant d'étendre ces résultats dans ces cas plus général.

De plus, dans la méthode Macro-Micro, une amélioration pourrait être apportée à la résolution du problème (**Macro**) qui tirerait parti de la structure de ce problème. En effet, comme discuté dans la section 3.2.3, la matrice hessienne associée au lagrangien de ce problème est une matrice creuse présentant une structure remarquable et qui peut être calculée simplement. Il est donc possible d'utiliser des méthodes d'optimisation du second ordre dans lesquelles l'exploitation de cette structure peut rendre la résolution du système linéaire associée à cette matrice hessienne très efficace.

Une troisième amélioration concerne la classe de fonctions utilisées pour approcher les fonctions valeur. Au lieu des *multilayer perceptrons* (MLPs), une possibilité serait en effet d'utiliser un autre type de réseaux de neurones, à savoir des *Physics-informed neural networks* (PINNs) [38]. Bien que l'architecture de ces réseaux de neurones soit similaire à celle des MLPs, la principale différence porte sur la fonction de perte utilisée dans l'apprentissage. Dans notre cas, cette fonction serait non plus l'erreur quadratique moyenne entre nos données et la prédiction de nos réseaux mais la somme pondérée de l'erreur quadratique moyenne à l'ordre 0 et à l'ordre 1. Un tel choix est d'autant plus intéressant que cette information du premier ordre est rendue disponible lors de la création de notre base de données section 5.2.1.1 à partir du co-état initial et final en utilisant le théorème 3.1.5. En plus de potentiellement améliorer l'initialisation de la méthode de tir simple des problèmes de commande optimale du problème (**Micro**) grâce là encore à l'utilisation du théorème 3.1.5, ces PINNs auraient probablement besoin de moins de données pour obtenir une précision semblable à celle des PMCs.

Ensuite, une caractéristique intéressante de la méthode de préconditionnement géométrique pourrait également être étudiée. En effet, il est possible que cette méthode introduise une invariance de la fonction de tir obtenue vis-à-vis des paramètres du problème. Il serait donc intéressant d'étudier l'influence de ces paramètres (temps final, condition initiale, ...) sur le problème (**E**) mais également l'influence du sous-intervalle

de temps pour le problème de répartition de couple. Cette invariance serait très intéressante puisqu'on obtiendrait toujours la même fonction de tir : une méthode de résolution spécifique de recherche de zéro de cette fonction de tir pourrait alors être développée qui fonctionnerait donc pour tous les paramètres possibles.

Finalement, une autre perspective serait de chercher d'autres formes d'ensemble accessible du système augmenté qui mènent à une fonction de tir linéaire. Plus spécifiquement, nous avons vu à travers l'exemple (E) que si la frontière de l'ensemble accessible correspond à un polynôme d'ordre 2, alors la fonction de tir utilisant la première méthode de normalisation est une fonction linéaire. Il serait donc possible d'appliquer la méthode de préconditionnement géométrique en approchant la frontière (ou une partie de la frontière) de l'ensemble accessible non pas par une ellipse mais par un polynôme du second ordre.





## Origine symplectique

---

L'objectif de cette annexe est de faire un petit détour sur l'origine de la dynamique hamiltonienne à partir de sa définition sur les variétés symplectiques. Nous nous intéressons plus précisément aux transformations canoniques (ou symplectomorphismes), c'est-à-dire aux changements de variables sur l'espace des phases qui conservent la dynamique hamiltonienne. Finalement, nous vérifierons que les relèvements donnés par la transformée de Mathieu sont des symplectomorphismes. La plupart des éléments de cette annexe sont bien connus dans la littérature [48]. Nous proposons cependant une nouvelle manière de prouver la proposition A.0.4, qui à notre connaissance est une humble contribution.

### Structure symplectique

On considère  $\mathbb{R}^n$  comme une variété différentielle de dimension  $n$ . L'espace tangent à  $\mathbb{R}^n$  en  $x \in \mathbb{R}^n$ , noté  $T_x\mathbb{R}^n$ , est construit à partir de l'ensemble des vecteurs tangents en  $x$  aux courbes de  $\mathbb{R}^n$ . On a donc  $T_x\mathbb{R}^n = \mathbb{R}^n$ . L'espace tangent à  $\mathbb{R}^n$  est défini par

$$T\mathbb{R}^n := \bigcup_{x \in \mathbb{R}^n} \{x\} \times T_x\mathbb{R}^n = \mathbb{R}^n \times \mathbb{R}^n.$$

L'espace co-tangent  $T_x^*\mathbb{R}^n$  (ou dual) de  $T_x\mathbb{R}^n$  en un point  $x \in \mathbb{R}^n$  est défini de manière analogue par l'ensemble des 1-formes sur  $T_x\mathbb{R}^n$ . L'espace  $T_x^*\mathbb{R}^n = (\mathbb{R}^n)^*$  est donc isomorphe à  $\mathbb{R}^n$ , dénoté  $T_x^*\mathbb{R}^n \simeq \mathbb{R}^n$ . Le fibré co-tangent est une variété différentielle de dimension  $2n$  et est défini de manière analogue par

$$T^*\mathbb{R}^n := \bigcup_{x \in \mathbb{R}^n} \{x\} \times T_x^*\mathbb{R}^n = \mathbb{R}^n \times (\mathbb{R}^n)^* \simeq \mathbb{R}^n \times \mathbb{R}^n.$$

La forme de Liouville  $\lambda: T^*\mathbb{R}^n \rightarrow T_z(T^*\mathbb{R}^n)$  est définie sur le fibré co-tangent par

$$\lambda(z) \cdot dz = p dx = \sum_{i=0}^n p_i dx_i.$$

pour tout  $dz = (dx, dp) \in T_z(T^*\mathbb{R}^n)$ , avec  $z = (x, p) \in T^*\mathbb{R}^n$ . La forme symplectique  $\omega: T_z(T^*\mathbb{R}^n) \times T_z(T^*\mathbb{R}^n) \rightarrow \mathbb{R}$  est définie à partir de la forme de Liouville par  $\omega = -d\lambda$  où  $d$  est l'opérateur de dérivée extérieure, c'est-à-dire

$$\omega = -d\lambda = -\sum_{i=0}^n dp_i \wedge dx_i = \sum_{i=0}^n dx_i \wedge dp_i,$$

avec  $\wedge$  le produit extérieur. On a donc pour tout  $z^k = (x^k, p^k) \in T^*\mathbb{R}^n$  et pour tout  $dz^k = (dx^k, dp^k) \in T_z(T^*\mathbb{R}^n)$  avec  $k = 1, 2$  :

$$\begin{aligned}\omega(dz^1, dz^2) &= \sum_{i=0}^n \begin{vmatrix} dx_i^1 & dp_i^1 \\ dx_i^2 & dp_i^2 \end{vmatrix} \\ &= \sum_{i=0}^n dx_i^1 dp_i^2 - dx_i^2 dp_i^1 \\ &= (dz^1 \mid \mathbf{J}_{2n} dz^2)\end{aligned}$$

On voit apparaître naturellement la matrice symplectique  $\mathbf{J}_{2n} \in \mathcal{M}_{2n, 2n}(\mathbb{R})$  définie par

$$\mathbf{J}_{2n} = \begin{bmatrix} \mathbf{0}_n & \mathbf{I}_n \\ -\mathbf{I}_n & \mathbf{0}_n \end{bmatrix},$$

et on remarque que  $\mathbf{J}_{2n}^{-1} = \mathbf{J}_{2n}^\top = -\mathbf{J}_{2n}$ , c'est-à-dire que cette matrice est antisymétrique et orthogonale. Le couple  $(T^*\mathbb{R}^n, \omega)$  est une variété symplectique, où  $\omega$  est la forme symplectique canonique de  $\mathbb{R}^n$ , c'est-à-dire une 2-forme différentielle fermée non dégénérée. Une fonction  $H$  réelle lisse définie sur  $T^*\mathbb{R}^n$  est appelée un hamiltonien, auquel on associe un champ de vecteur

$$\vec{H}: T^*\mathbb{R}^n \rightarrow T_z(T^*\mathbb{R}^n) \quad \text{par} \quad \omega(\vec{H}, \cdot) = dH$$

c'est-à-dire que pour tout  $z \in T^*\mathbb{R}^n$

$$\begin{aligned}\omega(\vec{H}(z), dz) &= dH(dz), & \forall dz \in T_z(T^*\mathbb{R}^n) \\ \iff (\vec{H}(z) \mid \mathbf{J}_n dz) &= (\nabla H(z) \mid dz), & \forall dz \in T_z(T^*\mathbb{R}^n) \\ \iff \mathbf{J}_n^\top \vec{H}(z) &= \nabla H(z) \\ \iff \vec{H}(z) &= \mathbf{J}_n \nabla H(z).\end{aligned}$$

On considère une famille d'hamiltonien  $\{H_t, t \in [t_0, t_f]\}$  et on propose de noter  $H(t, \cdot) = H_t(\cdot)$ . Le *flot hamiltonien* correspond à une famille de difféomorphismes  $\{\varphi_t, t \in [t_0, t_f]\}$  définis par

$$\frac{d}{dt} \varphi_t(z_0) = \vec{H}(t, \varphi_t(z_0)), \quad \varphi_{t_0}(z_0) = z_0.$$

Le flot hamiltonien correspond donc à la solution du problème de Cauchy

$$\dot{z}(t) = \vec{H}(t, z(t)), \quad z(t_0) = z_0, \quad \text{avec} \quad \vec{H}(t, z) = \mathbf{J}_n \nabla_z H(t, z).$$

On remarque que le champ de vecteur hamiltonien  $\vec{H}$  défini ici de manière canonique correspond quasiment à celui utilisé dans le cadre hamiltonien, Équation (2.18).

**Remarque.** On aurait exactement la même formulation si nous avions considéré le système augmenté présenté dans la section 4.1.1 et le cadre hamiltonien présenté dans la section 2.1.3.

Finalement, on remarque qu'il nous faut uniquement deux objets centraux : le hamiltonien  $H$  qui est une famille de fonctions réelles lisses définies sur le fibré co-tangent et la forme symplectique  $\omega$ . Ces deux éléments nous fournissent la dynamique du système en définissant canoniquement le champ de vecteur hamiltonien  $\vec{H}$ . Pour la suite de cette section, on considère un hamiltonien  $H$ , et non une famille, mais on garde en mémoire que les résultats obtenus peuvent aussi être appliqués dans le cadre non autonome.

## Transformation canonique

L'objectif est maintenant de savoir quels sont les changements de variables sur le fibré co-tangent qui conservent la dynamique hamiltonienne. Pour cela, nous allons définir les transformations canoniques, puis les caractériser à l'aide des matrices symplectiques, et montrer que ces transformations conservent la dynamique hamiltonienne.

### Définition A.0.1 – Transformation canonique

Un difféomorphisme  $\Phi: T^*\mathbb{R}^n \rightarrow T^*\mathbb{R}^n$  est un *symplectomorphisme* (ou transformation canonique, difféomorphisme symplectique) si et seulement si il préserve  $\omega$

$$\Phi^*\omega = \omega \iff \omega(\Phi'(z) \cdot u, \Phi'(z) \cdot v) = \omega(u, v), \quad \forall z \in T^*\mathbb{R}^n, \forall u, v \in T_z(T^*\mathbb{R}^n).$$

L'ensemble des matrices symplectiques est défini par

$$\mathcal{SP}_{2n}(\mathbb{R}) = \{A \in \mathcal{GL}_{2n}(\mathbb{R}) \mid A^\top \mathbf{J}_n A = \mathbf{J}_n\},$$

où  $\mathcal{GL}_{2n}(\mathbb{R})$  correspond à l'ensemble des matrices réelles inversibles de dimension  $2n$ . On remarque que  $A$  est une matrice symplectique si et seulement si  $A = -\mathbf{J}_n A^{-\top} \mathbf{J}_n$ . En effet :

$$A^\top \mathbf{J}_n A = \mathbf{J} \iff A = -\mathbf{J}_n A^{-\top} \mathbf{J}_n \quad (\text{A.1})$$

De plus,  $A^{-1}$  est symplectique :  $A^\top \mathbf{J}_n A = \mathbf{J}_n \iff \mathbf{J}_n = A^{-\top} \mathbf{J}_n A^{-1}$ .

### Proposition A.0.2 – Caractérisation des symplectomorphismes

Un difféomorphisme  $\Phi: T^*\mathbb{R}^n \rightarrow T^*\mathbb{R}^n$  est symplectomorphisme si et seulement si pour tout  $z \in T^*\mathbb{R}^n$ ,  $J_\Phi(z) \in \mathcal{SP}_{2n}(\mathbb{R})$ .

► Soit  $\Phi: T^*\mathbb{R}^n \rightarrow T^*\mathbb{R}^n$  un difféomorphisme, un point  $z \in T^*\mathbb{R}^n$  et deux éléments  $u, v \in T_z(T^*\mathbb{R}^n)$ . Si  $J_\Phi(z) \in \mathcal{SP}_{2n}(\mathbb{R})$ , c'est-à-dire que

$$J_\Phi(z)^\top \mathbf{J}_n J_\Phi(z) = \mathbf{J}_n,$$

alors

$$\begin{aligned} \omega(\Phi'(z) \cdot u, \Phi'(z) \cdot v) &= (J_\Phi(z) u \mid \mathbf{J}_n J_\Phi(z) v) \\ &= (u \mid J_\Phi(z)^\top \mathbf{J}_n J_\Phi(z) v) \\ &= (u \mid \mathbf{J}_n v) \\ &= \omega(u, v). \end{aligned}$$

Le difféomorphisme  $\Phi$  préserve donc  $\omega$ , c'est un symplectomorphisme. Réciproquement, si  $\Phi$  préserve  $\omega$ , on a alors

$$\begin{aligned} & \omega(\Phi'(z) \cdot u, \Phi'(z) \cdot v) = \omega(u, v) \\ \implies & (J_\Phi(z)u \mid \mathbf{J}_n J_\Phi(z)v) = (u \mid \mathbf{J}_n v) \\ \implies & u^\top J_\Phi(z)^\top \mathbf{J}_n J_\Phi(z)v = u^\top \mathbf{J}_n v \\ \implies & J_\Phi(z)^\top \mathbf{J}_n J_\Phi(z) = \mathbf{J}_n, \end{aligned}$$

c'est-à-dire que  $J_\Phi(z) \in \mathcal{SP}_{2n}(\mathbb{R})$ . ■

**Théorème A.0.3 – Conservation hamiltonienne [48, Théorème IV.A.1]**

*Les symplectomorphismes conservent la dynamique hamiltonienne, c'est-à-dire que si  $\vec{H}$  est un champ de vecteur associé à l'hamiltonien  $H$  et  $\Phi$  un symplectomorphisme, alors  $K = H \circ \Phi^{-1}$  est un hamiltonien et son champ de vecteur préserve les trajectoires : pour tout  $z \in T^*(\mathbb{R}^n)$ ,*

$$(\vec{K} \circ \Phi)(z) = J_\Phi(z)\vec{H}(z).$$

► Soit  $H$  un hamiltonien avec  $\vec{H}$  son champ de vecteur associé et  $\Phi$  une transformation canonique. Tout d'abord,  $K = H \circ \Phi^{-1}$  est une fonction réelle lisse définie sur  $T^*(\mathbb{R}^n)$ , c'est-à-dire un hamiltonien. On pose  $z \in T^*(\mathbb{R}^n)$  et  $y = \Phi(z) \in T^*(\mathbb{R}^n)$ . D'une part, on a

$$\nabla H(z) = J_\Phi(z)^\top (\nabla K \circ \Phi)(z) = J_\Phi(z)^\top \nabla K(y)$$

D'autre part,

$$\begin{aligned} \vec{K}(y) &= \mathbf{J}_n \nabla K(y) \\ &= \mathbf{J}_n J_\Phi(z)^{-\top} \nabla H(z) \\ &= -\mathbf{J}_n J_\Phi(z)^{-\top} \mathbf{J}_n \vec{H}(z) \\ &= J_\Phi(z) \vec{H}(z). \end{aligned} \quad \text{par (A.1)}$$

Comme nous venons de le voir avec le théorème A.0.3, être symplectique est une condition nécessaire pour qu'un difféomorphisme conserve la dynamique hamiltonienne. De plus, la proposition A.0.2 caractérise ces difféomorphismes par l'appartenance de leurs jacobiniennes à l'ensemble des matrices symplectiques. Cette dernière propriété est nécessaire pour définir la transformée de Mathieu. ■

## Transformée de Mathieu

L'objectif est maintenant de fournir un difféomorphisme qui conserve la dynamique hamiltonienne sur le fibré co-tangent, un symplectomorphisme, à partir d'un changement de variable sur l'espace d'état. On pose  $\phi: \mathbb{R}^n \rightarrow \mathbb{R}^n$  ce changement de variable, c'est-à-dire un difféomorphisme sur  $\mathbb{R}^n$ . On relève  $\phi$  sur le fibré co-tangent par le difféomorphisme  $\Phi: T^*\mathbb{R}^n \rightarrow T^*\mathbb{R}^n$  défini par

$$\Phi(x, p) = (\phi(x), J_\phi(x)^{-\top} p). \quad (\text{A.2})$$

Cette transformation est appelée la *transformée de Mathieu*.

### Proposition A.0.4 – Transformée de Mathieu

*La transformée de Mathieu est un symplectomorphisme.*

La preuve classique de ce résultat consiste à construire les quatre fonctions génératrices de symplectomorphismes et à définir la transformée de Mathieu à partir de l'une d'elle (c.f. [48, Chapitre 4.C]). Cependant, nous proposons de montrer cette proposition d'une autre façon qui nous semble plus simple et qui à notre connaissance est inédite.

► Soit  $\phi$  un difféomorphisme de  $\mathbb{R}^n$  et  $\Phi$  le relèvement de  $\phi$  sur le fibré co-tangent donné par la transformée de Mathieu (A.2). Pour montrer que  $\Phi$  est un symplectomorphisme, nous allons utiliser la caractérisation des symplectomorphismes donnée par la proposition A.0.2 et donc montrer que pour tout  $z \in T^*\mathbb{R}^n$ ,  $J_\Phi(z) \in \mathcal{SP}_{2n}(\mathbb{R})$ . Soit  $z = (x, p) \in T^*\mathbb{R}^n$ . La jacobienne de  $\Phi$  en  $z$  est donnée par

$$J_\Phi(z) = \begin{bmatrix} J_\phi(x) & \mathbf{0}_n \\ J_{F_p}(x) & J_\phi(x)^{-\top} \end{bmatrix} \quad \text{avec} \quad F_p: x \mapsto (J_\phi(x)^{-\top} p).$$

On doit maintenant montrer que cette matrice est symplectique, c'est-à-dire que  $J_\Phi(z)^\top \mathbf{J}_n J_\Phi(z) = \mathbf{J}_n$ . On a

$$\begin{aligned} J_\Phi(z)^\top \mathbf{J}_n J_\Phi(z) &= \begin{bmatrix} J_\phi(x)^\top & J_{F_p}(x)^\top \\ \mathbf{0}_n & J_\phi(x)^{-1} \end{bmatrix} \begin{bmatrix} \mathbf{0}_n & \mathbf{I}_n \\ -\mathbf{I}_n & \mathbf{0}_n \end{bmatrix} \begin{bmatrix} J_\phi(x) & \mathbf{0}_n \\ J_{F_p}(x) & J_\phi(x)^{-\top} \end{bmatrix} \\ &= \begin{bmatrix} J_\phi(x)^\top J_{F_p}(x) - (J_\phi(x)^\top J_{F_p}(x))^\top & \mathbf{I}_n \\ -\mathbf{I}_n & \mathbf{0}_n \end{bmatrix}. \end{aligned}$$

Il faut donc montrer que la matrice  $N(x, p) = J_\phi(x)^\top J_{F_p}(x)$  est une matrice symétrique. Pour cela, on pose la fonction  $f_p: \mathbb{R}^n \rightarrow \mathbb{R}$  définie par

$$f_p(y) = (\phi^{-1}(y) | p).$$

Le gradient de  $f_p$  en  $y$  est donc

$$\nabla f_p(y) = J_{\phi^{-1}}(y)^\top p = J_\phi(\phi^{-1}(y))^{-\top} p.$$

En posant  $y = \phi(x)$ , on remarque que  $\nabla f_p(\phi(x)) = F_p(x)$ . En notant  $\nabla^2 f_p(y)$  la matrice hessienne de la fonction  $f_p$  en  $y$ , on a donc  $J_{F_p}(x) = \nabla^2 f_p(\phi(x))J_\phi(x)$ . La hessienne est une matrice symétrique, ce qui implique que la matrice

$$N(x, p) = J_\phi(x)^\top \nabla^2 f_p(\phi(x))J_\phi(x)$$

est symétrique. Finalement, on a montré que  $J_\Phi(z)^\top \mathbf{J}_n J_\Phi(z) = \mathbf{J}_n$ , ou autrement dit que  $J_\Phi(z)$  est une matrice symplectique. La transformée de Mathieu  $\Phi$  d'un difféomorphisme  $\phi$  est donc bien un symplectomorphisme. ■

## Preuves du chapitre 2

---

### Section 2.3.2 : Lien avec le problème initial

#### Lemme B.0.1 – Bijection sur les points admissibles

*Le point  $(x, u)$  est admissible pour (OCP) si et seulement si le point  $\theta_p(x, u)$  est admissible pour (POCP).*

► Soit  $(x, u)$  un point admissible de (OCP). On pose  $(\bar{x}, \bar{u}) = \theta_p(x, u)$  et  $s \in [0, 1]$ . Par définition de  $\theta_p$ , et plus particulièrement celle de la fonction  $\mathbf{S}_{p,\cdot}$ , on a directement  $(\bar{x}, \bar{u}) \in \mathcal{D}_p$ . Ce point  $(\bar{x}, \bar{u})$  satisfait la dynamique d'état parallèle

$$\dot{\bar{x}}(s) = \begin{pmatrix} (\dot{x} \circ \phi_0)(s) \phi'_0(s) \\ \vdots \\ (\dot{x} \circ \phi_N)(s) \phi'_N(s) \end{pmatrix} = \begin{pmatrix} l_0 f(\phi_0(s), \bar{x}_0(s), \bar{u}_0(s)) \\ \vdots \\ l_N f(\phi_N(s), \bar{x}_N(s), \bar{u}_N(s)) \end{pmatrix} = f_p(s, \bar{x}(s), \bar{u}(s)),$$

les contraintes de contrôle parallèle

$$\bar{u}(s) = \begin{pmatrix} (u_0 \circ \phi_0)(s) \\ \vdots \\ (u_N \circ \phi_N)(s) \end{pmatrix} \in \begin{pmatrix} (U \circ \phi_0)(s) \\ \vdots \\ (U \circ \phi_N)(s) \end{pmatrix} = U_p(s),$$

ainsi que les contraintes d'état initiales et finales parallèles

$$c_p(\bar{x}(0), \bar{x}(1)) = \begin{pmatrix} \bar{x}_1(0) - \bar{x}_0(1) \\ \vdots \\ \bar{x}_N(0) - \bar{x}_{N-1}(1) \\ c(\bar{x}_0(0), \bar{x}_N(1)) \end{pmatrix} = \begin{pmatrix} x(t_1) - x(t_1) \\ \vdots \\ x(t_N) - x(t_N) \\ c(x(t_0), x(t_f)) \end{pmatrix} = 0.$$

Finalement si  $(x, u)$  est un point admissible de (OCP) alors  $\theta_p(x, u)$  est un point admissible de (POCP).

La réciproque est montrée de la même manière, en remarquant que pour tout point admissible  $(\bar{x}, \bar{u})$  de (POCP), il existe  $(x, u) \in \mathcal{D}$  tel que  $(\bar{x}, \bar{u}) = \theta_p(x, u)$ . Plus précisément, pour montrer que  $(x, u) \in \mathcal{D}$ , il faut

- utiliser l'appartenance de  $(\bar{x}, \bar{u})$  à  $\mathcal{D}_p$
- et utiliser les raccordements aux interfaces donnés par la fonction  $c_p$ .

■



**Lemme B.0.2 – Égalité des coûts**

Pour tout  $(x, u) \in \mathcal{D}$ , on a  $J(x, u) = J_p(\theta_p(x, u))$ .

► Soit  $(x, u) \in \mathcal{D}$ . On pose  $(\bar{x}, \bar{u}) = \theta_p(x, u)$  et on a

$$\begin{aligned}
 J_p(\bar{x}, \bar{u}) &= \int_0^1 f_p^0(s, \bar{x}(s), \bar{u}(s)) \, ds \\
 &= \int_0^1 \sum_{i=0}^N l_i f^0(\phi_i(s), \bar{x}_i(s), \bar{u}_i(s)) \, ds \\
 &= \sum_{i=0}^N \int_0^1 \phi_i'(s) f^0(\phi_i(s), (x \circ \phi_i)(s), (u \circ \phi_i)(s)) \, ds \\
 &= \sum_{i=0}^N \int_{t_i}^{t_{i+1}} f^0(t, x(t), u(t)) \, dt \\
 &= \int_{t_0}^{t_f} f^0(t, x(t), u(t)) \, dt \\
 &= J(x, u).
 \end{aligned}$$

■

## Section 2.3.4 : Lien entre les conditions nécessaires d'optimalité

### Lemme B.0.3 – Contraintes état/co-état

Pour tout vecteur  $a = (a_0, \dots, a_N)$  et  $b = (b_0, \dots, b_N)$  de  $(\mathbb{R}^{2n})^{N+1}$ , on a

$$g_p(a, b) \cong \begin{pmatrix} b_0 - a_1 \\ \vdots \\ b_{N-1} - a_N \\ g(a_0, b_N) \end{pmatrix}.$$

► Soit  $a, b \in (\mathbb{R}^{2n})^{N+1}$  et  $i \in \mathbb{N}_N$ . On propose d'utiliser les notations suivantes :

$$\begin{aligned} a &= (a^x, a^p) \in (\mathbb{R}^n)^{N+1} \times (\mathbb{R}^n)^{N+1}, & b &= (b^x, b^p) \in (\mathbb{R}^n)^{N+1} \times (\mathbb{R}^n)^{N+1}, \\ a^x &= (a_0^x, \dots, a_N^x) \in (\mathbb{R}^n)^{N+1}, & a^p &= (a_0^p, \dots, a_N^p) \in (\mathbb{R}^n)^{N+1}, \\ b^x &= (b_0^x, \dots, b_N^x) \in (\mathbb{R}^n)^{N+1}, & b^p &= (b_0^p, \dots, b_N^p) \in (\mathbb{R}^n)^{N+1}, \\ a_i &= (a_i^x, a_i^p) \in \mathbb{R}^{2n}, & b_i &= (b_i^x, b_i^p) \in \mathbb{R}^{2n}, \end{aligned}$$

$$\begin{aligned} h^x &= (a_1^x, \dots, a_N^x, b_0^x, \dots, b_{N-1}^x) \in (\mathbb{R}^n)^{2N}, \\ h^p &= (-a_1^p, \dots, -a_N^p, b_0^p, \dots, b_{N-1}^p) \in (\mathbb{R}^n)^{2N}, \\ h &= (h^x, h^p) \in (\mathbb{R}^n)^{4N}. \end{aligned}$$

Comme les fonctions impliquées dans la définition (2.26) de la fonction  $c_p$  n'ont aucune variable en commun, le noyau de  $c'_p$  peut être décomposé en deux noyaux, celui de  $c'$  et celui de  $d'$ . Autrement dit,

$$c_p^*(a, b) = \begin{pmatrix} d^*(h) \\ c^*(a_0, b_N) \end{pmatrix}$$

où  $d^*: (\mathbb{R}^n)^{4N} \rightarrow (\mathbb{R}^n)^{2N}$  est défini par  $d^*(h) = B_d(h^x)^\top h^p$ , et où  $B_d(h^x)$  est une matrice dont les colonnes forment une base de  $\text{Ker}(d'(h^x))$ . Plus précisément, on a

$$\text{Ker}(d'(h^x)) = \left\{ e \in (\mathbb{R}^n)^{2N} \mid d'(h^x)e = 0 \right\} = \left\{ e \in (\mathbb{R}^n)^{2N} \mid Le = 0 \right\},$$

et, en utilisant la définition de  $L$ , on peut prendre  $B_d(h^x)^\top = [\mathbf{I}_{nN} \ \mathbf{I}_{nN}] =: L^*$ . On a donc

$$d^*(h) = L^*h^p = \begin{pmatrix} b_0^p - a_1^p \\ \vdots \\ b_{N-1}^p - a_N^p \end{pmatrix}.$$

Donc pour tous éléments  $a, b \in (\mathbb{R}^{2n})^{N+1}$  :

$$g_p(a, b) = \begin{pmatrix} c_p^*(a, b) \\ c_p(a^x, b^x) \end{pmatrix} = \begin{pmatrix} d^*(h) \\ c^*(a_0, b_N) \\ d(h^x) \\ c(a_0^x, b_N^x) \end{pmatrix} \cong \begin{pmatrix} b_0 - a_1 \\ \vdots \\ b_{N-1} - a_N \\ g(a_0, b_N) \end{pmatrix}.$$

■

#### Lemme B.0.4 – Lien entre les pseudo-hamiltoniens

Pour tout temps  $s \in \mathbb{R}$ , pour tout couple état/co-état parallèle  $\bar{z} \in (\mathbb{R}^{2n})^{N+1}$  et pour tout contrôle parallèle  $\bar{u} \in (\mathbb{R}^m)^{N+1}$ , on a

$$h_p(s, \bar{z}, p^0, \bar{u}) = \sum_{i=0}^N l_i h(\phi_i(s), \bar{z}_i, p^0, \bar{u}_i).$$

► Soit  $s \in \mathbb{R}$ ,  $\bar{z} \in (\mathbb{R}^{2n})^{N+1}$  et  $\bar{u} \in (\mathbb{R}^m)^{N+1}$ . En utilisant la notation  $\bar{z}_i = (\bar{x}_i, \bar{p}_i)$ , on a directement

$$\begin{aligned} h_p(s, \bar{z}, p^0, \bar{u}) &= p^0 f_p^0(s, \bar{x}, \bar{u}) + (\bar{p} \mid f_p(s, \bar{x}, \bar{u})) \\ &= \sum_{i=0}^N \left( p^0 l_i f^0(\phi_i(s), \bar{x}_i, \bar{u}_i) + (\bar{p}_i \mid l_i f(\phi_i(s), \bar{x}_i, \bar{u}_i)) \right) \quad (\text{B.1}) \\ &= \sum_{i=0}^N l_i h(\phi_i(s), \bar{z}_i, p^0, \bar{u}_i). \end{aligned}$$

■

## Figures et tables

---

### Variables de l'application

Nom	Description	Unité
Variables intermédiaires		
$v$	Vitesse du véhicule	$\text{m.s}^{-1}$
$p$	Pente de la route	sans dimension
$\theta$	Angle entre la route et l'horizontale	rad
$a$	Accélération du véhicule	$\text{m.s}^{-2}$
$F_T$	Force totale appliquée	N
$F_r$	Force de résistance aérodynamique	N
$F_p$	Force de pesanteur	N
$F_d$	Force de propulsion	N
$T_{qF}$	Couple de freinage	N.m
$T_{qW_{ICE}}$	Part du couple aux roues généré par l'ICE	N.m
$T_{qW_{EM}}$	Part du couple aux roues généré par l'EM	N.m
$T_{qL_{ICE}}$	Pertes de couple de l'ICE (transmission)	N.m
$T_{qL_{EM}}$	Pertes de couple de l'EM (transmission)	N.m
$T_{qi_{ICE}}$	Couple idéal de l'ICE	N.m
$T_{qf_{ICE}}$	Pertes de frottements de l'ICE	N.m
$I$	Intensité du courant dans la batterie	A
$P_e$	Puissance électrique totale	W
$P_{EM}$	Puissance utilisée par l'EM	W
$P_T$	Puissance thermique totale	W
$P_f$	Puissance thermique due à la friction de l'ICE	W
$P_c$	Puissance thermique due à la combustion dans l'ICE	W
$P_a$	Puissance thermique transférée à l'air ambiant	W
$P_t$	Puissance thermique dégagée par le thermostat	W

TABLE C.1 – Liste des variables intermédiaires utilisées pour la description du modèle statique.

Nom	Description	Unité
Constantes		
$r$	Rayon des roues	m
$m$	Masse de la voiture	kg
$g$	Accélération de la pesanteur	$\text{m.s}^{-2}$
$P_e$	Rapport de transmission roues-EM	sans dimension
$P$	Rapport de transmission roues-ICE	sans dimension
$C$	Coefficient de conversion du couple en pression	$\text{m}^{-3}$
$Q$	Capacité équivalente de la batterie	A.s
$P_{add}$	Puissance électrique additionnelle moyenne	W
$C$	Capacité thermique du liquide de refroidissement	$\text{J.}^\circ\text{C}^{-1}$
$A$	Coefficient de modélisation	$\text{s}^{-1}$
$D$	Coefficient de transfert thermique entre l'ICE et l'air	$\text{W.}^\circ\text{C}^{-1}$
$T_a$	Température de l'air ambiant (extérieur)	$^\circ\text{C}$
$F$	Coefficient d'efficacité du thermostat	$\text{W.}^\circ\text{C}^{-1}$
$P_h$	Puissance thermique moyenne utilisée pour l'habitacle	W
$T_{qICE}^-$	Couple minimum de l'ICE	N.m
$T_{qICE}^+$	Couple maximum de l'ICE	N.m
$T_{qEM}^-$	Couple minimum de l'EM	N.m
$T_{qEM}^+$	Couple maximum de l'EM	N.m
$N_{ICE}^-$	Vitesse de rotation minimale de l'ICE	RPM
$N_{ICE}^+$	Vitesse de rotation maximale de l'ICE	RPM
$N_W^-$	Vitesse de rotation minimale des roues pour utiliser l'ICE	RPM
$SOC^-$	Borne minimale de l'état de charge de la batterie	sans dimension
$SOC^+$	Borne maximale de l'état de charge de la batterie	sans dimension

TABLE C.2 – Liste des constantes utilisées pour la description du modèle statique.

## BC-extrémale

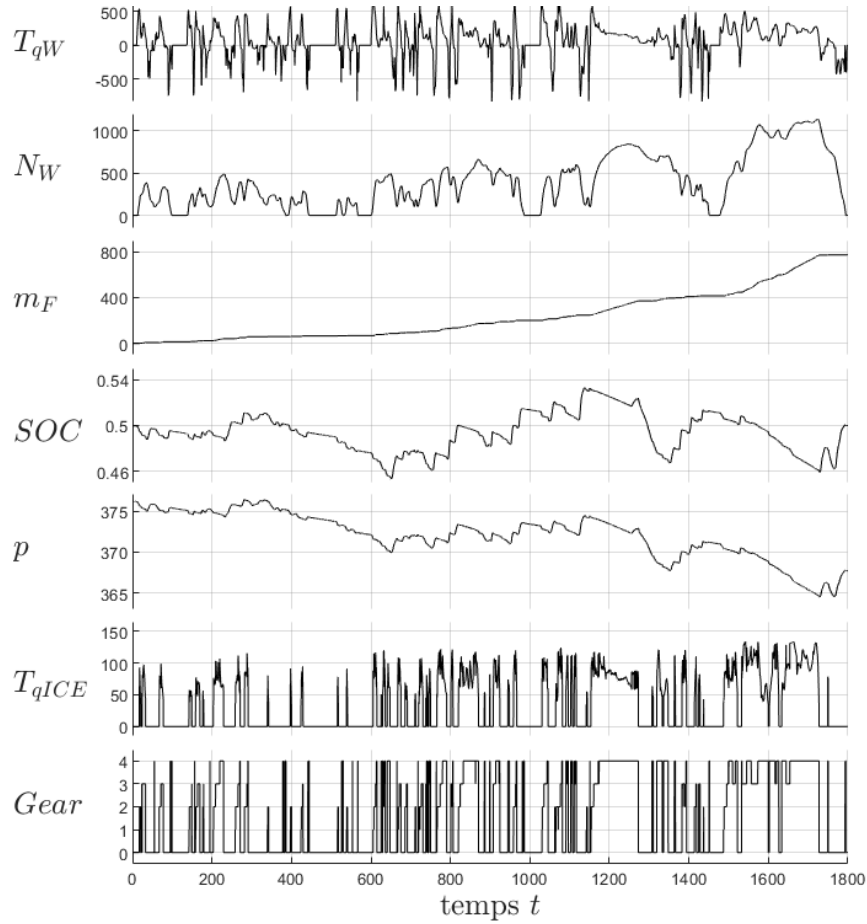


FIGURE C.1 – BC-extrémale  $(x, p, -1, u)$  associée au problème  $(P_3)$ , avec  $t_0 = 0$ ,  $t_f = 1800$  et  $x_0 = x_T = 0.5$ . L'état  $x$  est donné par  $SOC$ , le co-état correspond à  $p$  et la commande  $u$  correspond au couple  $(T_{qICE}, Gear)$ . Nous avons également affiché les deux séries temporelles issues du cycle WLTC, à savoir  $T_{qW}$  et  $N_w$ , ainsi que la trajectoire de coût  $m_F$ .

## Pénalisation

Dans le cas où la solution  $\tilde{X} = (\tilde{X}_1, \dots, \tilde{X}_N)$  du problème (5.2) n'est pas admissible, dans le sens où  $(x_0, \tilde{X}_1, \dots, \tilde{X}_N, x_T) \notin \mathcal{X}$ , il est possible d'approcher  $\mathcal{X}$  dans notre application. L'ensemble  $\mathcal{X}$  est défini par

$$\mathcal{X} = \left\{ (X_0, \dots, X_{N+1}) \in (\mathbb{R}^n)^{N+2} \mid (X_i, X_{i+1}) \in \Omega_i, \quad i \in \mathbb{N}_N \right\}. \quad (\text{C.1})$$

où  $\Omega_i$  correspond à l'ensemble des couples  $(a, b) \in \mathbb{R}^2$ , tels que l'état  $b$  est atteignable au temps  $t_{i+1}$  à partir de l'état  $a$  au temps  $t_i$ . Tout comme pour la fonction valeur, avoir une forme fermée de cet ensemble n'est pas trivial. Nous proposons donc de construire une approximation  $\tilde{\Omega}_i$  de  $\Omega_i$ , et d'utiliser l'équation (C.1) afin d'obtenir une approximation  $\tilde{\mathcal{X}}$  de  $\mathcal{X}$ .

À partir de la base de données  $\mathbb{D}_i$  préalablement calculée en section 5.2.1.1 avec la méthode 2, il est possible d'avoir une idée de la forme de  $\Omega_i$ , comme nous pouvons le voir dans la figure C.2.

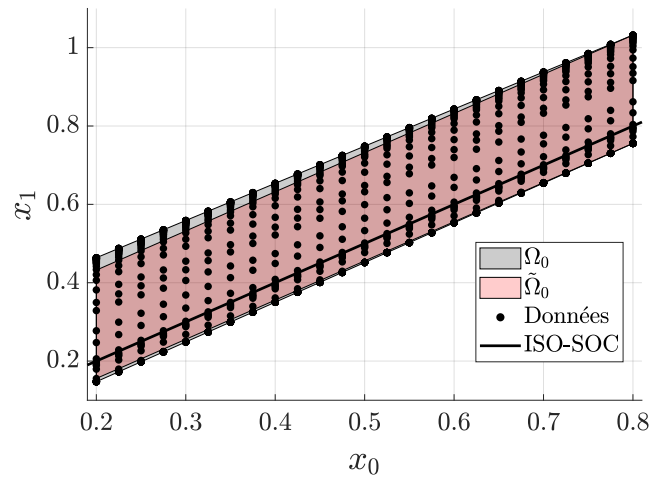


FIGURE C.2 – Évolution de l'état final  $x_1$  en fonction de l'état initial  $x_0$ . Les données proviennent de  $\mathbb{D}_0$ .

Nous avons choisi d'approcher cet ensemble par

$$\tilde{\Omega}_i = \left\{ (a, b) \mid b \in [a - \delta_i^-, a + \delta_i^+] \right\}$$

où  $\delta_i^-$  et  $\delta_i^+$  correspondent à deux scalaires positifs qui ne dépendent pas de  $x_0$ . Cette modélisation a été choisie pour sa simplicité, sa (relative) performance (c.f. Figure C.2) mais surtout pour son interprétation physique. En effet, pour un cycle donné, la puissance totale nécessaire pour la traction du véhicule est fixée. Il existe donc un nombre positif  $\delta_i^-$  tel que si toute cette puissance est fournie par le moteur électrique, on a à la fin du cycle  $x_f = x_0 - \delta_i^-$ . Ce même raisonnement peut être fait pour  $\delta_i^+$ . De plus, ces scalaires n'ont pas de raison physique de dépendre de l'état de charge initial de la voiture. Pour la détermination de ces deux valeurs,  $\delta_i^-$  et  $\delta_i^+$ , nous avons choisi de prendre

celles qui modélisent au mieux  $\Omega_i$  tout en nous assurant que  $\tilde{\Omega}_i \subset \Omega_i$ . Ce choix a été fait pour garantir que dans l'approximation  $\tilde{\Omega}_i$ , il n'y ait pas de couple d'état initial et final non admissibles. Nous avons donc calculé ces deux scalaires par

$$\delta_i^- = \min_{x_0 \in \mathbb{D}_i^-} \left\{ \max_{x_f \in \mathbb{D}_i^+(x_0)} \{x_0 - x_f\} \right\}, \quad \delta_i^+ = \min_{x_0 \in \mathbb{D}_i^-} \left\{ \max_{x_f \in \mathbb{D}_i^+(x_0)} \{x_f - x_0\} \right\},$$

avec  $\mathbb{D}_i^- = \{x_0 \mid (x_0, \cdot, \cdot) \in \mathbb{D}_i\}$  et  $\mathbb{D}_i^+(x_0) = \{x_f \mid (x_0, x_f, \cdot) \in \mathbb{D}_i\}$  qui correspondent respectivement aux ensembles d'état initial et final présents dans  $\mathbb{D}_i$ .

Finalement, il est possible de remplacer la contrainte  $X \in \mathcal{X}$  du problème (Macro) par la contrainte d'inégalité linéaire  $X \in \tilde{\mathcal{X}}$ , et d'utiliser des méthodes d'optimisation qui prennent en compte des contraintes linéaires d'inégalité, ou bien alors utiliser une méthode de pénalisation.



## Résultats de l'optimisation du problème (Macro)

Méthode	Temps Moyen	Minimum	Moyenne	Erreur Moyenne	Variance
Scénario : $x_0 = 0.3$ et $x_T = 0.7$ .					
NLopt MMA	0.3918	955.8035	961.0300	45.9410	13.5739
NLopt SLSQP	1.1901	915.0901	915.3762	0.2872	0.1131
NLopt DIRECT	36.4169	929.9577	929.9577	14.8687	0.0000
NLopt DIRECT-L	23.1629	928.9132	928.9132	13.8242	0.0000
NLopt CRS2-LM	87.9531	915.1113	<b>915.3027</b>	<b>0.2137</b>	0.0743
SP Nelder-Mead	10.4296	994.0043	1019.8629	104.7739	552.5540
SP Powell	19.5401	949.0289	958.0587	42.9697	16.1577
SP CG	0.3899	954.5008	959.3554	44.2664	18.1657
SP BFGS	0.4815	916.1381	945.9178	30.8288	846.0085
SP Newton-CG	<b>0.0119</b>	1135.9097	1179.1622	264.0732	667.4837
SP L-BFGS-B	0.4230	935.2914	940.0239	24.9349	16.1153
SP TNC	0.6056	955.9291	975.4533	60.3643	225.1188
SP COBYLA	2.0405	954.0280	961.8383	46.7493	63.5655
SP SLSQP	0.9954	<b>915.0890</b>	915.4450	0.3560	0.5144
SP trust-constr	6.1750	915.5087	915.8288	0.7398	0.0630
Scénario : $x_0 = 0.4$ et $x_T = 0.6$ .					
NLopt MMA	0.8461	877.7946	888.8955	51.6474	23.7246
NLopt SLSQP	1.1866	<b>837.2482</b>	838.3319	1.0837	4.8089
NLopt DIRECT	12.2379	845.6550	845.6550	8.4069	0.0000
NLopt DIRECT-L	28.3431	845.1093	845.1093	7.8611	0.0000
NLopt CRS2-LM	77.0756	837.4598	837.5820	0.3338	0.0173
SP Nelder-Mead	11.6488	915.6927	938.8815	101.6334	337.3047
SP Powell	25.7380	878.6298	882.1461	44.8980	4.3907
SP CG	0.5546	874.1088	879.8039	42.5558	23.7357
SP BFGS	0.5411	837.2632	857.4108	20.1627	541.9718
SP Newton-CG	<b>0.0132</b>	1061.3807	1099.8540	262.6058	768.4469
SP L-BFGS-B	0.2511	853.2174	867.2402	29.9921	78.4937
SP TNC	0.6241	884.6263	895.2542	58.0060	38.1445
SP COBYLA	2.3747	876.1365	888.3722	51.1240	164.8921
SP SLSQP	0.9992	837.3952	837.5979	0.3498	0.0237
SP trust-constr	7.1754	837.3607	<b>837.5589</b>	<b>0.3107</b>	0.0567
Scénario : $x_0 = 0.5$ et $x_T = 0.5$ .					
NLopt MMA	0.8729	805.8558	821.5535	55.1883	106.5946
NLopt SLSQP	1.1236	766.6189	768.5304	2.1651	1.4619
NLopt DIRECT	16.1301	769.3802	769.3802	3.0150	0.0000
NLopt DIRECT-L	24.6599	769.3092	769.3092	2.9439	0.0000
NLopt CRS2-LM	105.1026	766.4325	<b>766.6229</b>	<b>0.2576</b>	0.0204
SP Nelder-Mead	11.4202	855.9341	884.6465	118.2812	519.3839
SP Powell	28.3348	790.0941	808.3582	41.9929	70.6879
SP CG	0.6099	787.2092	804.4889	38.1236	131.6524
SP BFGS	0.4896	768.0073	789.5900	23.2247	1012.3094
SP Newton-CG	<b>0.0143</b>	985.0459	1031.0414	264.6761	822.2336
SP L-BFGS-B	0.3886	776.1688	780.9509	14.5857	12.2154
SP TNC	0.5768	819.8441	830.5608	64.1955	89.6949
SP COBYLA	2.2597	806.8076	821.1342	54.7689	351.8463
SP SLSQP	1.0017	767.0753	768.2115	1.8462	1.2047

SP trust-constr	8.4223	<b>766.3653</b>	766.7286	0.3633	0.1179
Scénario : $x_0 = 0.6$ et $x_T = 0.4$ .					
NLopt MMA	0.9075	719.0978	747.7087	51.4804	172.2114
NLopt SLSQP	1.2259	696.9079	699.5873	3.3590	3.3845
NLopt DIRECT	18.4782	703.5211	703.5211	7.2928	0.0000
NLopt DIRECT-L	30.9112	703.4749	703.4749	7.2466	0.0000
NLopt CRS2-LM	101.4881	<b>696.2283</b>	<b>696.5381</b>	<b>0.3098</b>	0.0663
SP Nelder-Mead	11.1195	801.3529	833.0471	136.8188	569.9389
SP Powell	28.0888	711.3960	729.3554	33.1271	129.5992
SP CG	0.8315	710.0226	720.5236	24.2953	44.7230
SP BFGS	0.4686	697.0555	715.6874	19.4591	862.5255
SP Newton-CG	<b>0.0128</b>	912.8818	972.9440	276.7157	1116.7475
SP L-BFGS-B	0.3425	702.3533	704.5496	8.3214	1.5364
SP TNC	0.6065	752.2634	771.1851	74.9569	328.3483
SP COBYLA	2.8323	730.5371	765.8650	69.6367	791.5268
SP SLSQP	1.0151	696.9034	699.3776	3.1493	3.2882
SP trust-constr	7.4325	696.4593	697.7454	1.5171	0.4767
Scénario : $x_0 = 0.7$ et $x_T = 0.3$ .					
NLopt MMA	0.9660	662.0759	676.1310	48.9516	42.8336
NLopt SLSQP	1.2081	627.5311	628.7661	1.5868	0.5209
NLopt DIRECT	16.4582	642.9242	642.9242	15.7449	0.0000
NLopt DIRECT-L	24.8531	642.5535	642.5535	15.3741	0.0000
NLopt CRS2-LM	120.8979	<b>627.1793</b>	<b>627.6348</b>	<b>0.4555</b>	0.0782
SP Nelder-Mead	11.7347	707.9197	763.1221	135.9428	1435.1385
SP Powell	33.4742	640.2566	655.4205	28.2412	127.0960
SP CG	0.7108	640.9360	658.5923	31.4129	180.1732
SP BFGS	0.4521	629.3118	657.3094	30.1301	1107.4854
SP Newton-CG	<b>0.0125</b>	863.6115	922.9559	295.7766	1169.9767
SP L-BFGS-B	0.3535	630.7828	632.1916	5.0123	0.5597
SP TNC	0.5112	687.2800	709.2932	82.1139	270.7473
SP COBYLA	3.0870	683.0822	723.4345	96.2552	1847.4220
SP SLSQP	1.1211	627.8010	628.7265	1.5472	0.7084
SP trust-constr	6.8688	627.2176	628.2259	1.0466	2.3023

TABLE C.3 – Comparaison entre plusieurs solveurs pour résoudre le problème (Macro) sur cinq scénarios (différentes conditions initiale et finale). Chacune des optimisations a été effectuée 10 fois avec une initialisation différente.

- La colonne « Methode » correspond au nom des méthodes utilisées dans la documentation du package associé.
- La colonne « Temps Moyen » correspond à la moyenne des temps de calcul sur les 10 optimisations.
- La colonne « Minimum » correspond à la plus petite évaluation du coût parmi les 10 solutions proposées.
- La colonne « Moyenne » correspond à la moyenne des évaluations du coût parmi les 10 solutions proposées.
- La colonne « Erreur moyenne » correspond à la moyenne de l'écart entre l'évaluation du coût et la plus petite valeur de la colonne « Minimum » parmi les 10 solutions proposées.
- La colonne « Variance » correspond à la variance des évaluations du coût parmi les 10 solutions proposées.



---

## Bibliographie

---

- [1] A. A. AGRACHEV AND Y. L. SACHKOV, *Control Theory from the Geometric Viewpoint*, Springer Berlin Heidelberg, 2004. [↔ 27, 28, 69 et 72.](#)
- [2] H. I. H. AZAMI, S. CAUX, F. MESSINE, M. SANS, AND S. KUTTER, *Probabilistic Optimal Control for Energy Management of Connected Hybrid Electrical Vehicles*, in 2018 International Symposium on Power Electronics, Electrical Drives, Automation and Motion (SPEEDAM), 2018. [↔ 16, 17 et 19.](#)
- [3] R. BELLMAN, *On the Theory of Dynamic Programming*, Proc. Natl. Acad. Sci., (1952). [↔ 19.](#)
- [4] H. BOCK AND K. PLITT, *A Multiple Shooting Algorithm for Direct Solution of Optimal Control Problems*, IFAC Proc. Vol., (1984). [↔ 33 et 62.](#)
- [5] T. J. BÖHME AND B. FRANK, *Hybrid Systems, Optimal Control and Hybrid Vehicles : Theory, Methods and Applications*, Springer International Publishing, 2017. [↔ 33.](#)
- [6] O. BOKANOWSKI, A. DÉSILLES, AND H. ZIDANI, *Relationship between maximum principle and dynamic programming in presence of intermediate and final state constraints*, ESAIM - Control Optim. Calc. Var., (2021). [↔ 53.](#)
- [7] V. G. BOLTYANSKII, *The Maximum Principle - How it came to be ?*, 1994. [↔ v, xi, 72 et 73.](#)
- [8] H. BORHAN, A. VAHIDI, A. M. PHILLIPS, M. L. KUANG, I. V. KOLMANOVSKY, AND S. DI CAIRANO, *MPC-Based Energy Management of a Power-Split Hybrid Electric Vehicle*, IEEE Trans. Control Syst. Technol., (2012). [↔ 16, 17 et 18.](#)
- [9] A. BOUALI, *Hybrid optimal control : optimality conditions and applications*, PhD thesis, Avignon Université, 2023. [↔ 13 et 15.](#)
- [10] A. BRAHMA, Y. GUEZENNEC, AND G. RIZZONI, *Optimal energy management in series hybrid electric vehicles*, in Proceedings of the 2000 American Control Conference, 2000. [↔ 16 et 19.](#)
- [11] A. E. BRYSON AND Y. C. HO, *Applied Optimal Control*, Taylor and Francis Group, 1975. [↔ 53.](#)
- [12] K.-H. N. BUI, J. CHO, AND H. YI, *Spatial-temporal graph neural network for traffic forecasting : An overview and open research issues*, Appl. Intell., (2021). [↔ 126.](#)
- [13] J.-B. CAILLAU, R. FERRETTI, E. TRÉLAT, AND H. ZIDANI, *An algorithmic guide for finite-dimensional optimal control problems*, in Handbook of numerical analysis : Numerical control, Part B, North-Holland, 2022. [↔ 18 et 51.](#)

- [14] A. CERNEA AND H. FRANKOWSKA, *A Connection Between the Maximum Principle and Dynamic Programming for Constrained Control Problems*, SIAM J. Control Optim., (2005). ↔ 53.
- [15] M. CERULLI, *Bilevel optimization and applications*, PhD thesis, Institut Polytechnique de Paris, 2021. ↔ 45.
- [16] L. CESARI, *Optimization—Theory and Applications : Problems with Ordinary Differential Equations*, Springer New York, 1983. ↔ 22 et 66.
- [17] Z. CHEN, Y. LIU, Y. ZHANG, Z. LEI, Z. CHEN, AND G. LI, *A neural network-based ecms for optimized energy management of plug-in hybrid electric vehicles*, Energy, (2022). ↔ 18 et 20.
- [18] Z. CHEN, C. C. MI, J. XU, X. GONG, AND C. YOU, *Energy Management for a Power-Split Plug-in Hybrid Electric Vehicle Based on Dynamic Programming and Neural Networks*, IEEE Trans. Veh. Technol., (2014). ↔ 19 et 20.
- [19] F. CLARKE, *Functional Analysis, Calculus of Variations and Optimal Control*, Springer London, 2013. ↔ 13, 15, 70, 75 et 78.
- [20] F. H. CLARKE AND R. B. VINTER, *The Relationship between the Maximum Principle and Dynamic Programming*, SIAM J. Control Optim., (1987). ↔ 53.
- [21] O. COTS, R. DUTTO, S. JAN, AND S. LAPORTE, *A bilevel optimal control method and application to the hybrid electric vehicle*, 2023. Submitted. ↔ 43, 93 et 131.
- [22] —, *Generation of value function data for bilevel optimal control and application to hybrid electric vehicle*, in Thematic Einstein Semester conference on Mathematical Optimization for Machine Learning, 2023. ↔ 93, 101 et 132.
- [23] —, *Geometric preconditioner for indirect shooting and application to hybrid vehicle*, in 4th IFAC Conference of Modelling, Identification and Control of Nonlinear Systems, 2024. ↔ 65 et 131.
- [24] E. CRISTIANI AND P. MARTINON, *Initialization of the Shooting Method via the Hamilton-Jacobi-Bellman Approach*, J. Optim. Theory Appl., (2010). ↔ 59.
- [25] S. DELPRAT, *Evaluation de stratégies de commande pour véhicules hybrides parallèles*, PhD thesis, Université de Valenciennes et du Hainaut-Cambresis, 2002. ↔ 17.
- [26] S. DEMPE AND A. ZEMKOHO, eds., *Bilevel Optimization*, Springer International Publishing, 2020. ↔ 46.
- [27] A. DMITRUK AND A. KAGANOVICH, *The Hybrid Maximum Principle is a consequence of Pontryagin Maximum Principle*, Syst. Control. Lett., (2008). ↔ 13.
- [28] P. DROZDZ AND N. FITZPATRICK, *A Hybrid Natural Gas Electric Fleet Vehicle*, in Future Transportation Technology Conference and Exposition, SAE International, 1996. ↔ 126.
- [29] A. DUBRAY, *Adaptation des lois de gestion d'énergie des véhicules hybrides suivant le profil de mission suivi*, PhD thesis, Institut National Polytechnique de Grenoble - INPG, 2002. ↔ 16 et 19.
- [30] R. V. GAMKRELIDZE, *History of the Discovery of the Pontryagin Maximum Principle*, Proceedings of the Steklov Institute of Mathematics, (2019). ↔ xi.

- [31] Q. GONG, Y. LI, AND Z. PENG, *Power management of plug-in hybrid electric vehicles using neural network based trip modeling*, in 2009 American Control Conference, 2009.  $\leftrightarrow$  126.
- [32] Q. GONG, Y. LI, AND Z.-R. PENG, *Optimal Power Management of Plug-in hybrid electric vehicles with trip modeling*, in ASME International Mechanical Engineering Congress and Exposition, 2007.  $\leftrightarrow$  19 et 126.
- [33] I. H. A. HAMZA, *Commande Prédicative optimale temps-réel, appliquée au contrôle de véhicules automobiles hybrides connectés à leurs environnements*, PhD thesis, INPT, 2018.  $\leftrightarrow$  16 et 18.
- [34] J.-B. HIRIART-URRUTY AND C. LEMARÉCHAL, *Fundamentals of Convex Analysis*, Springer Berlin Heidelberg, 2001.  $\leftrightarrow$  70.
- [35] J. HOFSTETTER, H. BAUER, W. LI, AND G. WACHTMEISTER, *Energy and Emission Management of Hybrid Electric Vehicles using Reinforcement Learning*, IFAC-Pap., (2019).  $\leftrightarrow$  17 et 19.
- [36] L. HYEOUN-DONG AND S.-K. SUL, *Fuzzy-logic-based torque control strategy for parallel-type hybrid electric vehicle*, IEEE Trans. Ind. Electron., (1998).  $\leftrightarrow$  18.
- [37] L. JOHANNESSON, M. ASBOGARD, AND B. EGARDT, *Assessing the potential of predictive control for hybrid vehicle powertrains using stochastic dynamic programming*, in Proceedings. 2005 IEEE Intelligent Transportation Systems, 2005.  $\leftrightarrow$  17, 19 et 125.
- [38] G. E. KARNIADAKIS, I. G. KEVREKIDIS, L. LU, P. PERDIKARIS, S. WANG, AND L. YANG, *Physics-informed machine learning*, Nat. Rev. Phys., (2021).  $\leftrightarrow$  132.
- [39] H. KAZEMI, Y. P. FALLAH, A. NIX, AND S. WAYNE, *Predictive AECMS by Utilization of Intelligent Transportation Systems for Hybrid Electric Vehicle Powertrain Control*, IEEE Trans. Intell. Veh., (2017).  $\leftrightarrow$  18.
- [40] M. KIM, D. JUNG, AND K. MIN, *Hybrid Thermostat Strategy for Enhancing Fuel Economy of Series Hybrid Intracity Bus*, IEEE Trans. Veh. Technol., (2014).  $\leftrightarrow$  18.
- [41] N. KIM, S. CHA, AND H. PENG, *Optimal Control of Hybrid Electric Vehicles Based on Pontryagin's Minimum Principle*, IEEE Trans. Control Syst. Technol., (2011).  $\leftrightarrow$  xii, 17 et 19.
- [42] A. B. KURZHANSKI AND P. VARAIYA, *Ellipsoidal techniques for reachability under state constraints*, SIAM Journal on Control and Optimization, 45 (2006).  $\leftrightarrow$  84.
- [43] L. S. PONTRYAGIN, V. G. BOLTYANSKII, R. V. GAMKRELIDZE, E. F. MISHECHENKO, *The Mathematical Theory of Optimal Processes*, Wiley Interscience, New York, 1962.  $\leftrightarrow$  xi, 19 et 23.
- [44] A. LE RHUN, *Stochastic optimal control for the energy management of hybrid electric vehicles under traffic constraints*, PhD thesis, Université Paris Saclay (COMUE), 2019.  $\leftrightarrow$  xii, 16, 17, 59 et 126.
- [45] C.-C. LIN, H. PENG, J. GRIZZLE, AND J.-M. KANG, *Power management strategy for a parallel hybrid electric truck*, IEEE Trans. Control Syst. Technol., (2003).  $\leftrightarrow$  16, 17 et 19.

- [46] M. M. SABRI, K. DANAPALASINGAM, AND M. RAHMAT, *A review on hybrid electric vehicles architecture and energy management strategies*, *Renew. Sust. Energ. Rev.*, (2016).  $\leftrightarrow$  15, 16 et 17.
- [47] A. A. MALIKOPOULOS, *Supervisory Power Management Control Algorithms for Hybrid Electric Vehicles : A Survey*, *IEEE Trans. Intell. Transp. Syst.*, (2014).  $\leftrightarrow$  17.
- [48] K. R. MEYER AND D. C. OFFIN, *Introduction to Hamiltonian Dynamical Systems and the N-Body Problem*, Springer International Publishing, 2017.  $\leftrightarrow$  135, 138 et 139.
- [49] Y. MILHAU, D. SINOQUET, AND G. ROUSSEAU, *Design optimization and optimal control for hybrid vehicles*, *Optim. Eng.*, (2011).  $\leftrightarrow$  17.
- [50] J. MORENO, M. ORTUZAR, AND J. DIXON, *Energy-management system for a hybrid electric vehicle, using ultracapacitors and neural networks*, *IEEE Trans. Ind. Electron.*, (2006).  $\leftrightarrow$  20.
- [51] C. MUSARDO, G. RIZZONI, Y. GUEZENNEC, AND B. STACCIA, *A-ECMS : An Adaptive Algorithm for Hybrid Electric Vehicle Energy Management*, *Eur. J. Control*, (2005).  $\leftrightarrow$  17 et 18.
- [52] V. NGO, T. HOFMAN, M. STEINBUCH, AND A. SERRARENS, *Optimal Control of the Gearshift Command for Hybrid Electric Vehicles*, *IEEE Trans. Veh. Technol.*, (2012).  $\leftrightarrow$  16, 17 et 19.
- [53] G. PAGANELLI, S. DELPRAT, T.-M. GUERRA, J. RIMAU, AND J.-J. SANTIN, *Equivalent consumption minimization strategy for parallel hybrid powertrains*, in *IEEE 55th Vehicular Technology Conference*, 2002.  $\leftrightarrow$  16, 17, 18 et 127.
- [54] A. PHILLIPS, M. JANKOVIC, AND K. BAILEY, *Vehicle system controller design for a hybrid electric vehicle*, in *Proceedings of the 2000. IEEE International Conference on Control Applications.*, 2000.  $\leftrightarrow$  18.
- [55] A. RAO, *A Survey of Numerical Methods for Optimal Control*, *Adv. Astronaut. Sci.*, (2010).  $\leftrightarrow$  33.
- [56] A. RAO AND K. MEASE, *Dichotomic basis approach to solving hyper-sensitive optimal control problems*, *Automatica*, (1999).  $\leftrightarrow$  33.
- [57] A. RAUH, A. BOURGOIS, L. JAULIN, AND J. KERSTEN, *Ellipsoidal enclosure techniques for a verified simulation of initial value problems for ordinary differential equations*, in *2021 International Conference on Control, Automation and Diagnosis (ICCAD)*, IEEE, 2021.  $\leftrightarrow$  84.
- [58] G. ROUSSEAU, *Véhicule hybride et commande optimale*, PhD thesis, ENMP, 2008.  $\leftrightarrow$  19.
- [59] F. R. SALMASI, *Control Strategies for Hybrid Electric Vehicles : Evolution, Classification, Comparison, and Future Trends*, *IEEE Trans. Veh. Technol.*, (2007).  $\leftrightarrow$  18.
- [60] H. SCHÄTTLER AND U. LEDZEWICZ, *Geometric optimal control : theory, methods and examples*, Springer, 2012.  $\leftrightarrow$  53 et 72.

- [61] J. SCORDIA, M. DESBOIS-RENAUDIN, R. TRIGUI, B. JEANNERET, F. BADIN, AND C. PLASSE, *Global optimisation of energy management laws in hybrid vehicles using dynamic programming*, Int. J. Veh. Des., (2005).  $\leftrightarrow$  16, 17 et 19.
- [62] L. SERRAO, S. ONORI, AND G. RIZZONI, *A Comparative Analysis of Energy Management Strategies for Hybrid Electric Vehicles*, J. Dyn. Syst. Meas. Control, (2011).  $\leftrightarrow$  18 et 128.
- [63] M. SIVERTSSON AND L. ERIKSSON, *Design and Evaluation of Energy Management using Map-Based ECMS for the PHEV Benchmark*, Oil Gas Sci. Technol. - Rev. IFP Energies nouvelles, (2015).  $\leftrightarrow$  18.
- [64] J. STOER AND R. BULIRSCH, *Introduction to Numerical Analysis*, Springer New York, 2002.  $\leftrightarrow$  33.
- [65] R. S. SUTTON, *Learning to predict by the methods of temporal differences*, Mach. Learn., (1988).  $\leftrightarrow$  19.
- [66] R. S. SUTTON AND A. G. BARTO, *Reinforcement Learning : An Introduction*, The MIT Press, 2018.  $\leftrightarrow$  19.
- [67] E. TATE AND S. BOYD, *Finding Ultimate Limits of Performance for Hybrid Electric Vehicles*, SAE Trans., (2000).  $\leftrightarrow$  18.
- [68] H. TIAN, S. LI, X. WANG, Y. HUANG, AND G. TIAN, *Data-driven hierarchical control for online energy management of plug-in hybrid electric city bus*, Energy, (2017).  $\leftrightarrow$  20.
- [69] E. TRÉLAT, *Contrôle optimal : théorie & applications*, Vuibert, 2005.  $\leftrightarrow$  23 et 74.
- [70] E. TRÉLAT, *Contrôle optimal*, 2008. Notes de cours, Master de Mathématiques, Université d'Orléans.  $\leftrightarrow$  18.
- [71] V. M. VELIOV, *Lipschitz continuity of the value function in optimal control*, Journal of Optimization Theory and Applications, 94 (1997).  $\leftrightarrow$  132.
- [72] X. WANG, H. HE, F. SUN, AND J. ZHANG, *Application Study on the Dynamic Programming Algorithm for Energy Management of Plug-in Hybrid Electric Vehicles*, Energies, (2015).  $\leftrightarrow$  xii, 17 et 19.
- [73] C. WENG, Y. WANG, V. TSOURAPAS, C. PATIL, AND J. SUN, *Optimal control of hybrid electric vehicles with power split and torque split strategies : A comparative case study*, in Proceedings of the 2011 American Control Conference, 2011.  $\leftrightarrow$  17.
- [74] S. G. WIRASINGHA AND A. EMADI, *Classification and Review of Control Strategies for Plug-In Hybrid Electric Vehicles*, IEEE Trans. Veh. Technol., (2011).  $\leftrightarrow$  17.
- [75] S. XIE, X. HU, S. QI, AND K. LANG, *An Artificial Neural Network-Enhanced Energy Management Strategy for Plug-In Hybrid Electric Vehicles*, Energy, (2018).  $\leftrightarrow$  18, 20 et 128.
- [76] Y. ZHANG, L. CHU, Z. FU, N. XU, C. GUO, D. ZHAO, Y. OU, AND L. XU, *Energy management strategy for plug-in hybrid electric vehicle integrated with vehicle-environment cooperation control*, Energy, (2020).  $\leftrightarrow$  17, 18 et 126.
- [77] J. ZHAO AND J. WANG, *Integrated Model Predictive Control of Hybrid Electric Vehicles Coupled With Aftertreatment Systems*, IEEE Trans. Veh. Technol., (2016).  $\leftrightarrow$  17 et 18.







**Titre :** Méthode à deux niveaux et préconditionnement géométrique en contrôle optimal. Application au problème de répartition de couple des véhicules hybrides électriques.

**Mots clés :** Contrôle optimal à deux niveaux, Préconditionnement géométrique, Méthodes indirectes, Principe du maximum de Pontryagin, Apprentissage Machine, Véhicule hybride

**Résumé :** Motivé par le problème industriel de répartition de couple dans les véhicules hybrides électriques, ce travail propose principalement deux nouvelles méthodes de résolution indirectes de problèmes de commande optimale. La première est la méthode Macro-Micro qui est basée sur une décomposition à deux niveaux du problème de commande optimale, faisant intervenir les fonctions valeur de Bellman de manière explicite à des temps préalablement fixés. Ces fonctions sont connues pour être assez difficile à construire. L'idée principale est d'approcher ces fonctions valeur par des réseaux de neurones, ce qui mène à une résolution hiérarchique d'un problème d'optimisation en dimension faible et d'un ensemble de problèmes de commande optimale définis sur des intervalles de temps plus courts. La seconde est une méthode de préconditionnement géométrique qui permet une résolution plus efficace du problème de commande optimale. Cette méthode, basée sur l'interprétation géométrique du co-état et sur la transformée de Mathieu, utilise un changement de variable linéaire à partir de la simple transformation d'une ellipse en cercle. Ces deux méthodes, bien que présentées séparément, peuvent être combinées et mènent à une résolution plus rapide, robuste et légère du problème de répartition de couple, permettant ainsi que de s'approcher des critères d'embarquabilités.

**Title:** Bilevel method and geometric preconditioner in optimal control. Application to the torque split and gear shift optimal control problem.

**Key words:** Bilevel optimal control, Geometric preconditioner, Indirect methods, Pontryagin maximum principle, Machine learning, Hybrid vehicle

**Abstract:** Motivated by the torque split and gear shift industrial problem of hybrid electric vehicles, this work mainly proposes two new indirect optimal control problem methods. The first one is the Macro-Micro method, which is based on a bilevel decomposition of the optimal control problem and uses Bellman's value functions at fixed times. These functions are known to be difficult to create. The main idea of this method is to approximate these functions by neural networks, which leads to a hierarchical resolution of a low dimensional optimization problem and a set of independent optimal control problems defined on smaller time intervals. The second one is a geometric preconditioning method, which allows a more efficient resolution of the optimal control problem. This method is based on a geometrical interpretation of the Pontryagin's co-state and on the Mathieu transformation, and uses a linear diffeomorphism which transforms an ellipse into a circle. These two methods, presented separately, can be combined and lead together to a fast, robust and light resolution for the torque split and gear shift optimal control problem, closer to the embedded requirements.