Stochastic optimal control for the energy management of hybrid electric vehicles under traffic constraints

Arthur Le Rhun

NNT : 2019SACLX084

THÈSE DE DOCTORAT

de

L'UNIVERSITÉ PARIS-SACLAY

École doctorale de mathématiques Hadamard (EDMH, ED 574)

Établissement d'inscription : École polytechnique

Établissement d'accueil : IFP Énergies nouvelles

Laboratoire d'accueil : Centre de mathématiques appliquées de Polytechnique, UMR 7641 CNRS

Spécialité de doctorat : Mathématiques appliquées

Arthur LE RHUN

Stochastic Optimal Control for the Energy Management of Hybrid Electric Vehicles under Traffic Constraints

Date de soutenance : 12 Décembre 2019 *à* Palaiseau

Après avis des rapporteurs : NADIR FARHI (IFSTTAR)
NICOLAS PETIT (Mines ParisTech)

Jury de soutenance :

FRÉDÉRIC BONNANS	(INRIA) Directeur de thèse
JEAN PHILIPPE CHANCELIER	(École des Ponts ParisTech) Examinateur
GIOVANNI DE NUNZIO	(IFP Énergies nouvelles) Invité
NADIR FARHI	(IFSTTAR) Rapporteur
STÉPHANE GAUBERT	(INRIA) Président
THOMAS LEROY	(IFP Énergies nouvelles) Examinateur
PIERRE MARTINON	(INRIA) Codirecteur de thèse
NICOLAS PETIT	(Mines ParisTech) Rapporteur

To my parents,

Remerciements

Je tiens tout d'abord à remercier mes directeurs de thèse et mes encadrants pour le temps passé et la patience dépensée durant ces trois années. Tout d'abord, Frédéric Bonnans pour m'avoir donné l'opportunité d'effectuer cette thèse sous sa direction. Ainsi que Pierre Martinon pour la finesse de son analyse numérique et ses encouragements dans la dernière ligne droite. Thomas Leroy pour ses connaissances sur le domaine automobile et sa rigueur dans la communication des résultats. Giovanni De Nunzio pour sa présence et sa disponibilité malgré la distance et l'éloignement. Je tiens à les remercier pour leur confiance et leur ouverture d'esprit par rapport aux approches proposées.

Je remercie également Nadir Farhi et Nicolas Petit d'avoir bien voulu être mes rapporteurs et de leur intérêt pour mes travaux, ainsi que Jean Philippe Chancelier et Stéphane Gaubert d'avoir accepter de faire partie du jury de cette thèse.

Je me dois aussi de remercier Iadine Chadés, Sam Nicol, et Yann Dujardin d'avoir accompagner mes premiers pas dans le monde de la recherche au sein du CSIRO, de m'avoir donné envie de faire un doctorat ainsi que de m'avoir encouragé à explorer de nouveaux domaines et de nouvelles voies. Je tiens aussi à remercier l'ensemble des enseignants du M2 Optimization de Paris-Saclay, tout particulièrement Pierre Carpentier pour avoir été mon tuteur à l'ENSTA, il m'a fait découvrir le domaine de l'optimisation et a su me guider dans mes choix et orientations.

Si j'ai pu mener cette thèse, c'est aussi grâce aux équipes administratives du département R11 à l'IFPEN, Carole et Francoise, et aussi de l'équipe COMMANDS à l'INRIA, Jessica et Hanadi, je vous remercie vivement de m'avoir aider dans mes démarches et demandes.

Je remercie ce qui on fait que chaque jour de travail, fut aussi un moment de convivialité. Mes collègues de l'IFPEN, Bruno, Oliver, Vincent, Elsie, Elodie, pour ces repas un plus longs que la moyenne mais si agréables. Les doctorants du bureau 2010, Heythem pour ses histoires, Fedor et Martin pour leur passion des échecs, Céline pour avoir mis l'ambiance dans un bureau si consciencieux, Juliette pour avoir tant remarqué mon absence. Merci tout particulièrement à toi, Antoine, de m'avoir supporté depuis mon stage de fin d'études et pendant toute ma thèse, pour ces promenades du déjeuner et de leur conversations plus personnelles, et de ces si nombreuse places d'opéra durement obtenue au shotgun. Je tiens à féliciter tous ceux qui ont soutenu avant moi pour leur travail et leurs efforts, et à souhaiter bonne chance aux autres : le bout du chemin n'est plus si loin.

Si cette thèse est beaucoup de travail, je tiens aussi à remercier ceux qui ont su m'en sortir. Tout d'abord par le sport, merci à Romain de m'avoir laissé sortir ma frustration sur la piste, et à José, Thibaut, et Flora d'en avoir subi physiquement les conséquences. Merci à Denis, Yoko pour les conversations certes toujours animées mais toujours intéressantes, à Philippine pour la relecture, et de m'avoir subi pendant la rédaction, et aussi aux habitués de la Kfet d'avoir rendu les mercredis soir comme un bol d'air dans la semaine. Je remercie également Axel pour sa joie et son inépuisable énergie. Je tiens aussi à remercier particulièrement mes amis qui ont su me supporter pendant si longtemps, Paul-Claude, Corentin, Marine, malgré mon caractère. Ainsi que Chaan qui malgré sa fuite dans le Sud est toujours là pour s'amuser.

Je finirai par ceux qui ont fait le plus pour cette thèse, mes parents. Merci d'être là depuis 26 ans et de m'avoir toujours encouragé.

Preface

Contents

Abstract

The focus of this PhD thesis is to design an optimal Energy Management System (EMS) for a Hybrid Electric Vehicle (HEV) following traffic constraints. In the current state of the art, EMS are typically divided between real-time designs relying on local optimization methods, and global optimization that is only suitable for off-line use due to computational constraints. The starting point of the thesis is that in terms of energy consumption, the stochastic aspect of the traffic conditions can be accurately modelled thanks to (speed,acceleration) probability distributions. In order to reduce the data size of the model, we use clustering techniques based on the Wasserstein distance, the corresponding barycenters being computed by either a Sinkhorn or Stochastic Alternate Gradient method. Thanks to this stochastic traffic model, an off-line optimization can be performed to determine the optimal control (electric motor torque) that minimizes the fuel consumption of the HEV over a certain road segment. Then, a bi-level algorithm takes advantage of this information to optimize the consumption over a whole travel, the upper level optimization being deterministic and therefore fast enough for real-time implementation. We illustrate the relevance of the traffic model and the bi-level optimization, using both traffic data generated by a simulator, as well as some actual traffic data recorded near Lyon (France). Finally, we investigate the extension of the bi-level algorithm to the eco-routing problem, using an augmented graph to track the state of charge information over the road network.

Résumé

Cette thèse aborde la conception d'un Système de Gestion Énergétique (EMS), prenant en compte les contraintes de trafic, pour un véhicule hybride électrique. Actuellement, les EMS sont habituellement classés en deux catégories: ceux proposant une architecture en temps réel cherchant un optimum local, et ceux qui recherchent un optimum global, plus coûteux en temps de calcul et donc plus approprié à un usage hors ligne. Cette thèse repose sur le fait que la consommation énergétique peut être modélisée précisément à l'aide de distributions de probabilité sur la vitesse et l'accélération. Dans le but de réduire la taille des données, une classification est proposée, basée sur la distance de Wasserstein, les barycentres des classes pouvant être calculés grâce aux itérations de Sinkhorn ou la méthode du Gradient Stochastique Alterné. Cette modélisation du trafic permet à une optimisation hors ligne de déterminer le contrôle optimal (le couple du moteur électrique) qui minimise la consommation de carburant du véhicule hybride sur un segment routier. Un algorithme bi-niveau tire avantage de cette information afin d'optimiser la consommation sur l'ensemble du trajet. Le niveau supérieur d'optimisation étant déterministe, il est suffisamment rapide pour une implémentation en temps réel. La pertinence du modèle de trafic et de la méthode bi-niveau est illustrée à l'aide de données trafic générées par un simulateur, mais aussi grâce à des données réelles collectées près de Lyon (France). Enfin, une extension de la méthode bi-niveau au problème d'éco-routage est présentée, utilisant un graphe augmenté pour déterminer l'état de charge lors du chemin optimal.

Dissertation Outline

1. Introduction

The introduction presents the context that motivated the thesis, and recalls the state of the art in Energy Management Systems for Hybrid Electric Vehicles.

2. A stochastic data-based traffic model

A new approach to estimate the energy consumption of a vehicle based on traffic conditions is proposed, reducing traffic data to (speed,acceleration) probability distributions. The reduction is done over a subdivision of the road network into small segments induced by the network topography. In order to reduce data occupancy, clustering techniques are applied, allowing different times of the day to be grouped together in a single class of traffic conditions. Two methods (average and memoryless sampling) are proposed to estimate the energy consumption of a vehicle based on this traffic model. The relevance of both methods is investigated using data generated by a traffic simulator, with a sensitivity analysis with respect to different parameters such as speed/acceleration discretization, timeframe length and number of clusters. Finally, a real-life scenario using floating car data is studied as well, in order to check the applicability and robustness of the proposed method.

3. An Optimal local policy using traffic prediction

In this chapter, a way to obtain optimal policies that minimize the fuel consumption while taking into account the traffic conditions is presented. A key point is that we assume the speed and acceleration of the vehicle to follow the probability distributions of the traffic stochastic model. Working at the 'micro' scale of individual road segments, we present a Stochastic Dynamic Programming formulation to compute the optimal electric torque, for given initial and final states of charge. Solving off-line a collection of these 'micro' problems provides a cost map of the consumption depending on the entry and exit state of charge, for all road segments and associated traffic conditions. The value functions can be used to recover the corresponding optimal control (electric torque) in a feedback form as well.

4. An Optimal State of charge trajectory on a travel with traffic conditions

Here, we formulate the optimization problem at the upper ('macro') level for the whole travel. The main idea is to compute an optimal sequence of target state of charge at the end of all successive road segments. Using the knowledge database described in chapter 3 (cost maps), the problem results in resolving by Dynamic Programming. This 'macro' optimization framework provides both the state of charge trajectory and optimal policy along the travel, and is fast enough for on-line implementation. The approach is validated by numerical simulations based on actual traffic data.

5. Application to the optimal routing problem under traffic conditions

An extension of the bi-level formulation to the eco-routing problem is proposed. Using the knowledge of traffic conditions over the entire road network, we search both the optimal path and state of charge trajectory. This problem results in finding the shortest path on a weighted graph whose nodes are (position, state of charge) pairs for the vehicle, the edge cost being evaluated thanks to the cost maps from the 'micro' optimizations. The classical A^* algorithm is used to solve the problem, with a heuristic based on a lower bound of the energy needed to complete the travel. The eco-routing method is validated by numerical simulations and compared to the fastest path on a synthetic road network.

6. Conclusions

The main results of the thesis are recalled here, and several perspectives are presented.

Appendix A. A Stochastic gradient method for Wasserstein barycenters of distributions

This appendix presents a method to compute barycenters of probability distributions, in the sense of the distance of Wasserstein, related to optimal transport. We introduce an algorithm based on stochastic gradient, that solves an entropic regularization of the barycenter problem. We prove the convergence rate and complexity of the algorithm, and compare it to the classical Sinkhorn iterations method.

List of Publications

Journals

[56] A. Le Rhun, F. Bonnans, G. De Nunzio, T. Leroy, and P. Martinon, "A Stochastic Data-Based Traffic Model Applied to Vehicles Energy Consumption Estimation," IEEE Transactions on Intelligent Transportation Systems, pp. 1–10, 2019.

Preprints

[55] A. Le Rhun, F. Bonnans, G. De Nunzio, T. Leroy, and P. Martinon, "A bi-level energy management strategy for HEVs under probabilistic traffic conditions," preprint, Sept. 2019, URL: https:// hal.inria.fr/hal-02278777. (Submitted in IEEE Transactions on Control Systems and Technology.)

[57] A. Le Rhun, F. Bonnans, G. De Nunzio, T. Leroy, and P. Martinon, "An Ecorouting algorithm for HEVs under traffic conditions," preprint, Nov. 2019, URL: https://hal.inria.fr/hal-02356277. (Submitted to the 21st IFAC World Congress.)

Résumé de la Thèse

Les contributions exposées dans cette thèse sont doubles :

- **Un modèle stochastique de trafic**, utilisant une classification des distributions de probabilités de (vitesse,accélération), capable d'estimer de façon efficace la distribution de consommation énergétique des véhicules sur un segment routier, avec une occupation mémoire réduite.
- **Un EMS bi-niveau** intégrant l'aléa dû aux conditions de trafic, calculant en temps réel la trajectoire optimale de la charge de la batterie et du couple électrique.

Dans la suite, les contributions et les résultats de chaque chapitre sont présentés.

Chapitre 2 - Un modèle stochastique de trafic

Une nouvelle approche, pour estimer la consommation énergétique du trafic, est proposée. Elle repose sur l'agrégation des données trafic en distributions de probabilité (vitesse/accélération), voir Fig.1. Une telle agrégation peut être faite sur chaque segment composant le réseau routier.

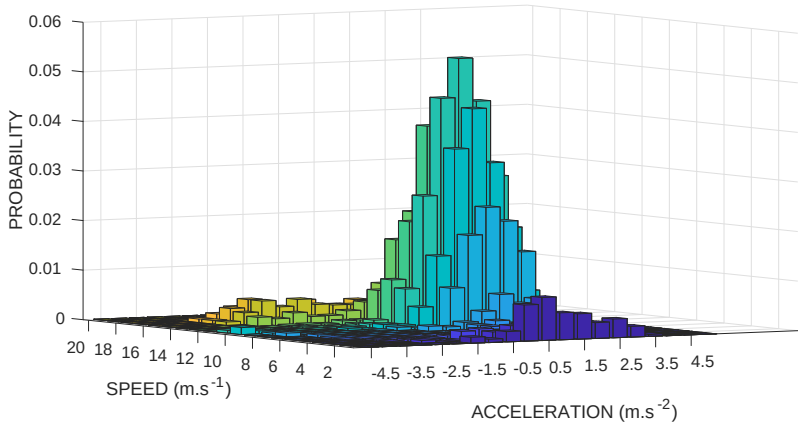


Figure 1 – Exemple de distribution de trafic

Dans le but de réduire la taille des données générées, des techniques de classification ont été utilisées afin d'obtenir des classes de conditions de trafic représentatives. La distance utilisée, lors de la classification, est la distance de Wasserstein, basée sur la théorie du transport optimal :

$$W_1(\mu,\nu) = \min_{\pi\in\Pi(\mu,\nu)} d\cdot\pi \qquad (0.0.0.1)$$

où $\Pi(\mu, \nu)$ est l'ensemble des plans de transports de μ à ν, i.e. l'ensemble des matrices positives π de taille $(|N_{\mathbb{V}}| \times |N_{\mathbb{A}}|)^2$ avec comme marginales $\pi^\top \mathbf{1} = \mu$ et $\pi \mathbf{1} = \nu$. Les barycentres, obtenus grâce à cette distance, gardent les aspects géométriques des distributions initiales de probabilités . Différentes heures de la journée possédant des conditions de trafic similaires sont ainsi groupées dans la même classe, voir Fig.2.

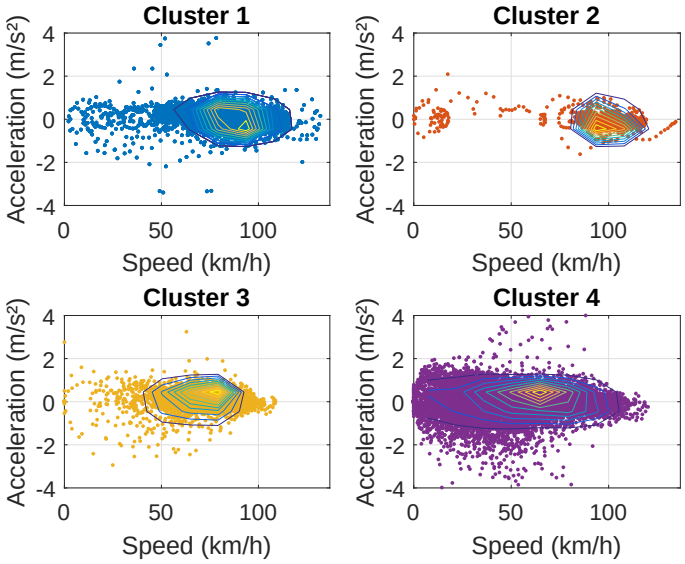


Figure 2 – Illustration des classes de trafic obtenues avec leurs données brutes.

Différents modèles de consommation énergétique, basés sur ces distributions, sont proposés afin d'estimer la consommation des véhicules sur le réseau routier.

Le premier modèle moyenne la puissance selon la distribution du trafic $\bar{\mu}$:

$$\bar{P}(\bar{\mu}) = \sum_{\mathbb{V}\times\mathbb{A}} \bar{\mu}_t(v, a) P(v, a) \tag{0.0.0.2}$$

La puissance moyenne est ensuite intégrée sur l'intervalle de temps $[t_i, t_f]$ passé par le véhicule sur le segment :

$$C_{Avg} = \int_{t_i}^{t_f} \bar{P}(\bar{\mu}_t) dt \tag{0.0.0.3}$$

La seconde méthode repose sur l'idée que les véhicules doivent suivre le trafic, dans un sens statistique. La méthode génère donc une suite de (vitesse/accélération) qui est intégrée pour obtenir la consommation. Cette suite est tirée à partir de la distribution de trafic $\bar{\mu}$.

Afin de valider les méthodes, un simulateur de trafic a été utilisé pour générer des données et comparer les consommations énergétiques estimées à celles mesurées. Une analyse approfondie de la sensibilité des méthodes par rapport au paramètres (i.e. nombre de classes, taille des supports des distributions, etc.) a été menée dans le cadre des ces données simulées, voir Fig.3. Pour finir, un scénario réel est analysé afin d'évaluer l'applicabilité et la robustesse de la méthode.

Les résultats montrent que la classification permet de réduire la taille des données (jusqu'à 99%) tout gardant suffisamment d'information pour estimer la distribution des consommations énergétiques.

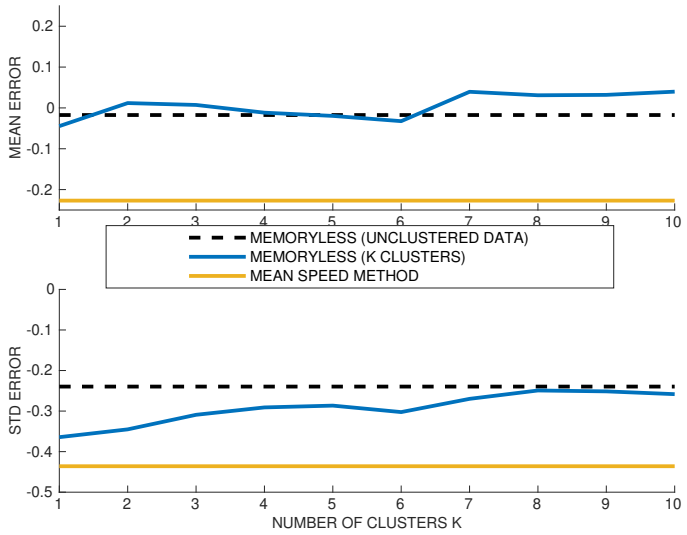


Figure 3 – Erreurs dues à la classification.

Chapitre 3 - Une stratégie optimale locale grâce à la prédiction du trafic

Une méthode pour obtenir une stratégie optimale malgré les aléas du trafic est présentée. L'hypothèse principale est que la vitesse et l'accélération du véhicule $(\mathbf{V}, \mathbf{A})$ suivent la distribution de trafic selon un modèle i.i.d.. De telles distributions ont été obtenues dans le chapitre 2.

Le trafic est ainsi pris en compte afin d'obtenir le couple optimal du moteur électrique T_m satisfaisant la demande de charge du véhicule. Le contrôle minimisant la consommation de carburant C du moteur thermique selon une cible finale d'état de charge de la batterie SoC_f et la carte de consommation électrique du moteur thermique P_m. La discrétisation de ce problème de contrôle stochastique optimal sur un segment de longueur L_s peut être écrite comme suit :

$$\min_{T_m} \quad \mathbb{E}\left[\sum_{k=0}^{k_f} h^k C(\mathbf{V}^k, \mathbf{A}^k, T_m^k,) + P(SoC_f, SoC^{k_f})\right] \tag{0.0.0.4}$$

$$s.c\ \forall k, \quad SoC^{k+1} = SoC^k + \frac{h^k}{C_{max}} P_m(\mathbf{V}^k, \mathbf{A}^k, T_m^k) \tag{0.0.0.5}$$

$$D^{k+1} = D^k + h^k \mathbf{V}^k \tag{0.0.0.6}$$

$$T_m^k \in [T_{min}, T_{max}] \tag{0.0.0.7}$$

$$SoC^k \in [0, 1] \tag{0.0.0.8}$$

$$h^k := \min(h_0, (L_s - D^k)/\mathbf{V}^k) \tag{0.0.0.9}$$

$$k_f := 1 + \max\{k, D^k + h_0 \mathbf{V}^k < L_s\} \tag{0.0.0.10}$$

$$SoC^0 = SoC_0;\ D^0 = 0. \tag{0.0.0.11}$$

Résoudre ce problème pour l'ensemble des états de charge initiaux possibles permet d'obtenir l'espérance de consommation selon l'état de charge final ainsi qu'une stratégie optimale capable de la réaliser, voir Fig.4.

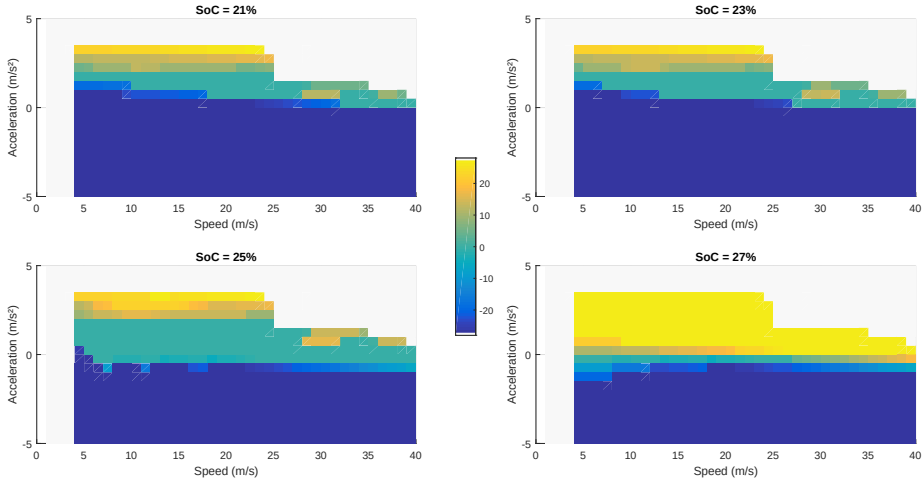


Figure 4 – Couple optimal du moteur électrique dans le cas d'une circulation fluide.

Les erreurs dues à la discrétisation du problème et à la résolution numérique ont été étudiées.

Les simulations numériques sur des données réelles, collectées sur l'autoroute A7 située près de Lyon, ont été comparées à une approche déterministe. Celle-ci donne le minimum de consommation atteignable pour un véhicule si son profil de vitesse est connu. Les résultats indiquent que la méthode proposée dans ce chapitre parvient à traiter l'aléa dû aux conditions de circulation avec une surconsommation raisonnable, avec un écart d'environ 5% à la borne théorique, voir Fig.5.

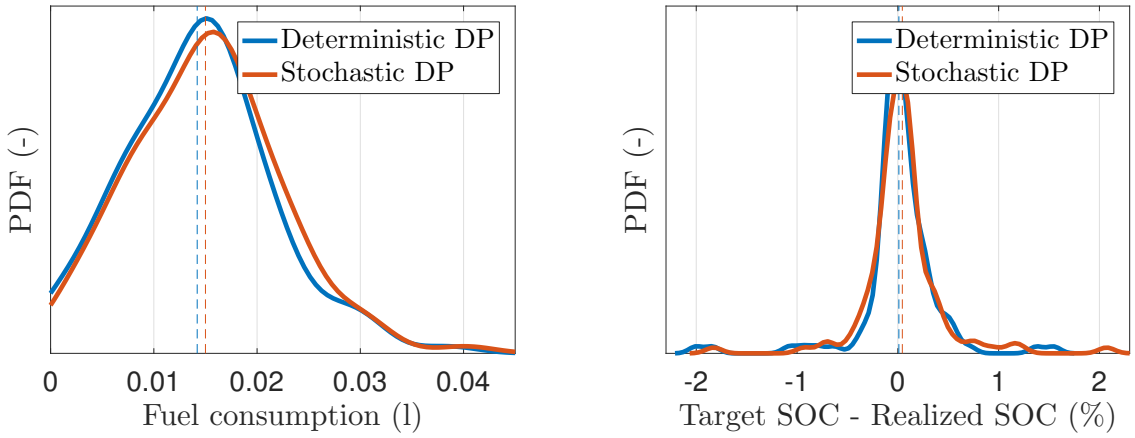


Figure 5 – Comparaison entre la méthode déterministe et la méthode stochastique

Chapitre 4 - Une trajectoire optimale d'état de batterie sur un trajet avec prévision de trafic

En raison du coût numérique de la programmation dynamique stochastique, il est possible de résoudre celle-ci sur de petits segments routiers, mais des longueurs de trajets plus grandes empêchent son implémentation en temps réel. Afin d'obtenir en temps réel des stratégies optimales adaptées aux trajets, une méthode bi-niveau, basée sur une décomposition spatiale, est proposée.

La méthode du chapitre 3 calcule le couple optimal du moteur électrique sur un segment s connaissant les conditions de trafic, représentées par une distribution de probabilité μ_s. Cette programmation dynamique peut être effectuée hors ligne sur les classes de trafic obtenues au chapitre 2. Grâce à la résolution hors de ces problèmes, des cartes de coût $\mathcal{C}$ dépendant de l'état de charge initial et final peuvent être calculées et archivées, voir Fig.6.

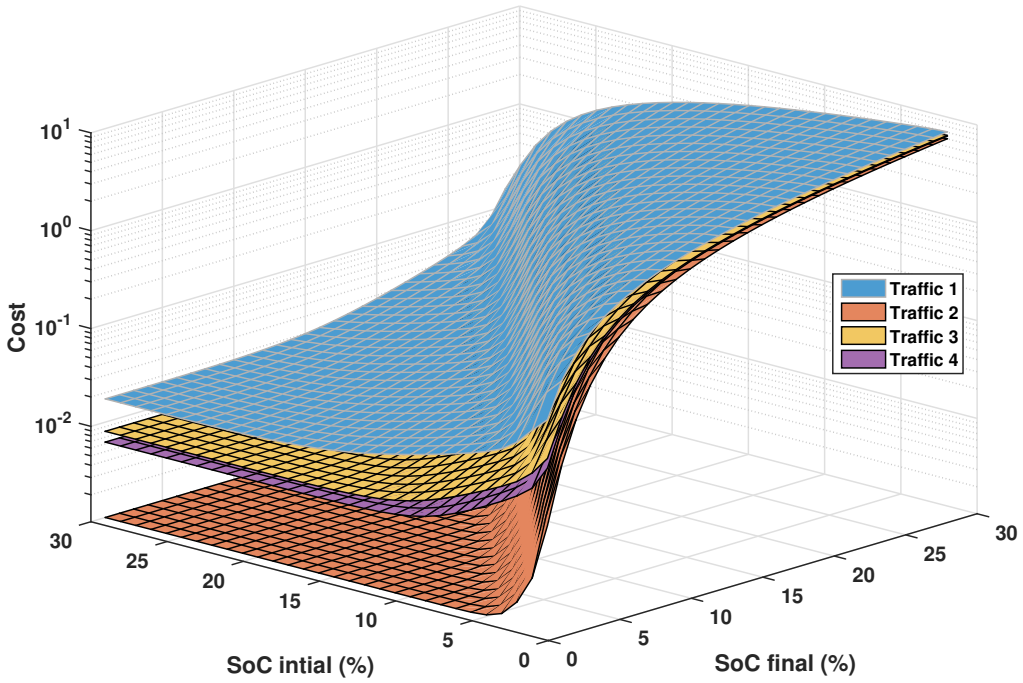


Figure 6 – Exemple de carte de coût.

Ces cartes de coût permettent de prédire le coût moyen d'un segment en fonction de l'état de charge au début et à la fin du segment. Par conséquent, un nouveau problème qui calcule une référence pour l'état de charge SoC^r à atteindre à la fin de chaque segment, en fonction de ces cartes de coût, est défini ainsi :

$$\min_{SoC^r} \quad \sum_{s=0}^{S-1} \mathcal{C}(SoC_s^r, SoC_{s+1}^r, \mu_s) \tag{0.0.0.12}$$

$$s.t.\ \forall s, \quad SoC_s^r \in [0,1] \tag{0.0.0.13}$$

$$SoC_0^r = SoC_{intial} \tag{0.0.0.14}$$

$$SoC_S^r = SoC_{final} \tag{0.0.0.15}$$

Le principal avantage de cette méthode bi-niveau est que l'incertitude du comportement du trafic a déjà été traitée grâce aux optimisations stochastiques proposées dans le chapitre 3. Ainsi, le nouveau problème peut être résolu grâce à une programmation dynamique déterministe et peut donc être calculé en ligne. La trajectoire de l'état de charge est obtenue tout au long du parcours ainsi que la stratégie optimale sur chaque segment.

Des simulations numériques ont été effectuées afin de comparer la consommation obtenue avec une borne inférieure théorique, calculée grâce à la programmation dynamique déterministe sur les profils de vitesse. Les résultats montrent que la méthode bi-niveau introduit une surconsommation ($\sim 15\%$) et satisfait bien la contrainte d'état de charge final, tout en maintenant un temps de simulation inférieur à 10 secondes.

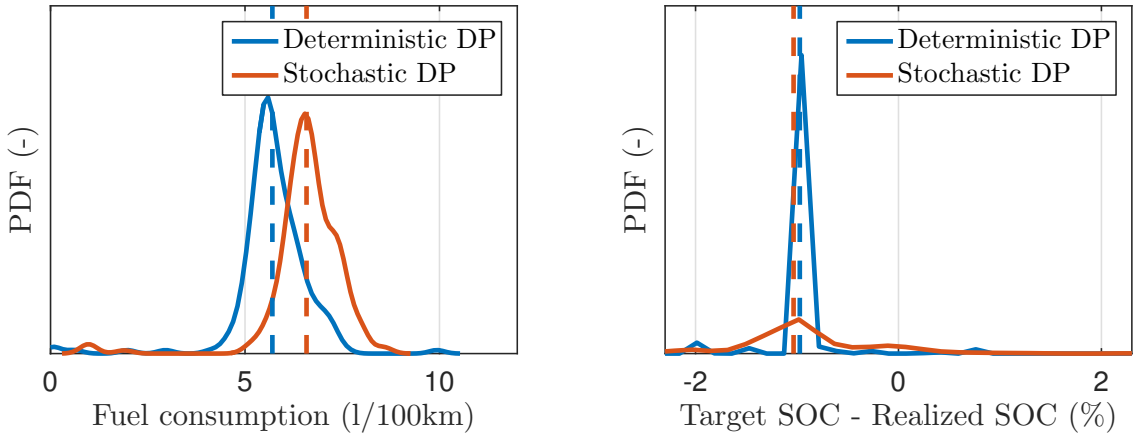


Figure 7 – Comparaison entre la méthode déterministe et la méthode bi-niveau. Conditions $SoC_0 = 0.25, SoC_f = 0.24$.

Chapitre 5 - Application au problème de routage optimal sous contrainte de trafic

Sur les réseaux routiers, plusieurs chemins sont possibles pour arriver à la destination souhaitée. Les conditions de circulation sur ces chemins peuvent varier considérablement, par exemple en raison de limitations de vitesse ou d'embouteillages. Afin de minimiser la consommation globale, nous proposons d'utiliser les méthodes développées dans les chapitres précédents pour trouver le chemin de consommation optimale, en fonction de la consommation et des conditions de circulation, dans le réseau routier.

Ce dernier est modélisé sous forme de graphe, où chaque intersection correspond à un nœud. Ce graphe est étendu dans chaque nœud avec l'état de charge possible, afin de calculer à la fois le chemin optimal et la trajectoire optimale de l'état de charge. L'ensemble de tous les chemins possibles entre l'origine S et la destination D est noté Γ_{SD}. Le problème du plus court chemin s'écrit alors comme suit :

$$\underset{\gamma \in \Gamma_{SD}}{\text{minimize}} \quad \sum_{i \in \gamma} \nu_{N_i N_{i+1}}(SoC^r_{i+1}, SoC^r_i) \tag{0.0.0.16}$$

$$\text{s.t} \quad \forall i \in \gamma, SoC^r_i \in [0,1] \tag{0.0.0.17}$$

$$\forall i \in \gamma, SoC^r_{i+1} - SoC^r_i \in [-a_i, b_i] \tag{0.0.0.18}$$

L'erreur due à la discrétisation de l'état de charge est analysée. Un algorithme classique A^* est utilisé pour résoudre le problème. Les simulations numériques ont montré que la consommation d'un tel chemin est réduite par rapport au chemin le plus rapide et que les contraintes SoC_f sont relativement satisfaites.

Annexe A - Méthode du gradient stochastique pour les barycentres de Wasserstein

Dans cette annexe, un nouvel algorithme est présenté, basé sur des méthodes de gradient stochastique pour calculer la distribution de probabilités du barycentre de Wasserstein. La distance de Wasserstein, basée sur le transport optimal, compare deux distributions. En désignant μ le barycentre des distributions ν^k, le plan de transport π est calculé comme suit :

$$\begin{aligned} &\min_{\pi,\mu} \sum_{k=1}^{K} c^k \cdot \pi^k; \;\; \pi \geq 0; \\ &(\pi^k)^\top \mathbf{1} = \mu, \;\; \pi^k \mathbf{1} = \nu^k, \;\; k = 1, \ldots, K, \end{aligned} \tag{0.0.0.19}$$

Des méthodes rapides de calcul de ces barycentres sont nécessaires lors du traitement de grandes quantités de données. Grâce à une double régularisation entropique du problème et en utilisant la forme semi-duale du problème régularisé par l'entropie inverse, nous obtenons un problème qui peut être résolu grâce à un algorithme de la classe du gradient stochastique :

$$\max_{u} \sum_{j} h^j(u) \tag{0.0.0.20}$$

avec :

$$h^j(u) = -\varepsilon_1 \sum_{k} \nu_j^k (\log(\frac{1}{\nu_j^k} \sum_{i \in X^0} \xi_{ij}^k e^{\frac{-u_i^k}{\varepsilon_1}}) + 1) - \frac{\varepsilon_2 \sum_k \nu_j^k}{K} (\sum_{i \in X^0} e^{\frac{\sum_{r=1}^{K} u_i^r}{\varepsilon_2}}) \tag{0.0.0.21}$$

La complexité temporelle et spatiale de l'algorithme a été comparée aux itérations Sinkhorn pour les barycentres introduites en 2015 , voir [8].

Table of Contents

List of Figures

List of Tables

List of Algorithms

CHAPTER 1

Introduction

Contents

1.1 Global context

Nowadays, the world faces great changes and issues. This thesis tries to provide solutions to some of these issues thanks to the technological evolution of the world. The improvement of the technology made possible the creation of intelligent vehicles and their hybridization to reduce the environmental impact due to individual cars greenhouse gas emission. This section recalls the issues and how the current technological evolutions can help to find new solutions.

1.1.1 Environmental issues

Since the beginning of the 70s, the number of cars has more than tripled, see Fig.1.1. This leads to an increased fuel consumption and therefore an increased atmospheric pollution, which is one of the primary cause of early mortality according to WHO, see [3].

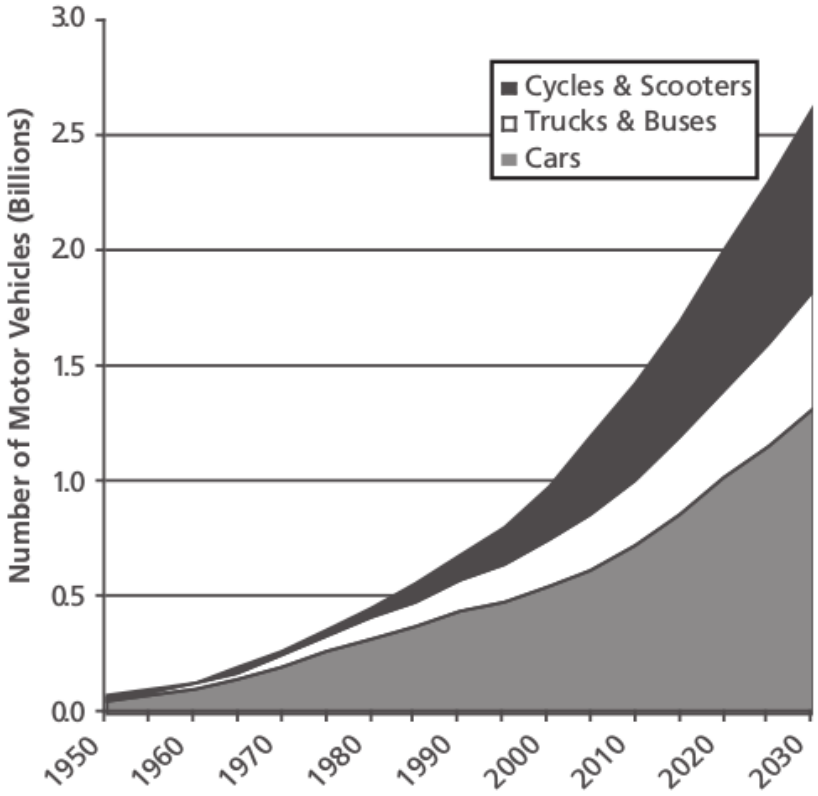


Figure 1.1 – Historical and projected worldwide number of vehicles *Source:[83]*

The International Energy Agency ([46]) also estimates that the current trends in energy use and supply are unsustainable economically, environmentally and socially. Without decisive actions, energy-related emissions of carbon dioxide (CO_2) will more than double by 2050, and increased oil demand raises concerns over the security of supplies. The IEA also estimated that over 50% of oil used around the world is for transportation, and three-quarters of the energy used in the transportation sector is consumed on the roads. Therefore, it seems crucial that governments around the world tackle the problem of vehicle fuel economy.

In the EU case, CO_2 emission targets have been set for the newly registered vehicles. These targets are defined as a percentage reduction from the 2021 levels and impose a reduction of 15% by 2025 and 37.5% by 2030 [33].

1.1.2 Hybridization

In order to reach the fleet-wide emission targets, the automotive industry tries to develop vehicles using alternate sources of energy. This approach allows to reduce the consumption of vehicles, and therefore the oil dependence and the environmental issue. Since the beginning of the 21^{st} century, there has been a strong development of electric (EV) and hybrid electric vehicles (HEV). Such vehicles use, totally or partially, electricity for propulsion. HEV in particulare are well suited for the trade-off between reduction of fuel consumption and operating range. Indeed, the use of the electric motor allows to target the most efficient operating points of the internal combustion engine.

Sales of such vehicles are now non negligeable and tend to increase, see Fig.1.2. Market share projections seem to agree to this tendency, predicting that purely electric vehicles (EVs) will reach at least 8% of all vehicle sales by 2025, while hybrid electric vehicles (HEVs) will rise to 23% of market share [41, 27]. Therefore, optimizing the fuel consumption of these vehicles will become an increasingly important matter.

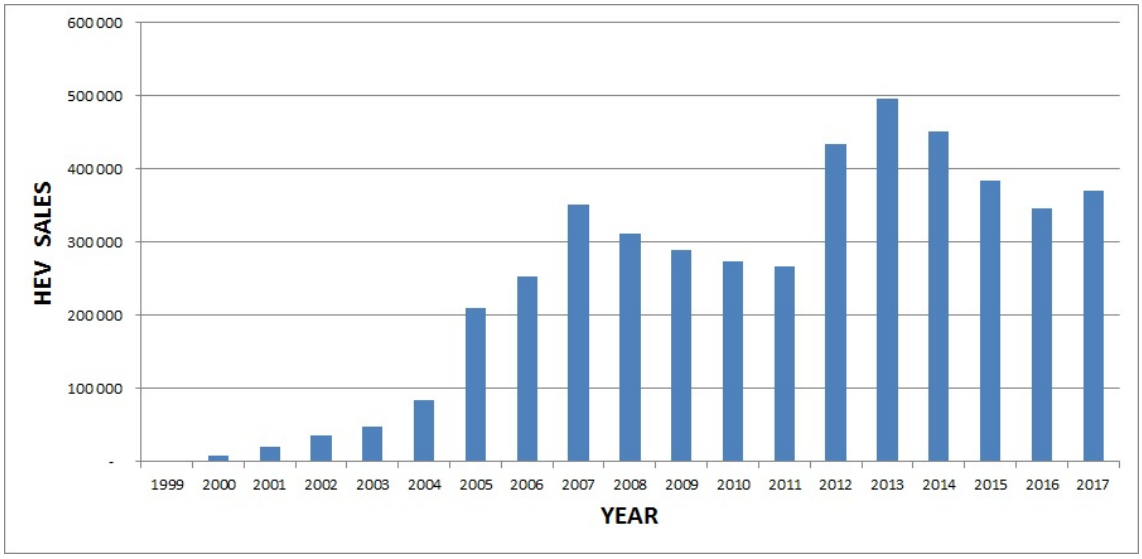


Figure 1.2 – Sales of HEV on the US market *Source: U.S. Department of Energy*

1.1.3 Connected world

Since the beginning of the 50s, electronics devices have become more and more present in modern societies, and cars do not escape to this tendancy. This leads to a fundamental change in vehicle control systems, namely the advent of Intelligent Transportation Systems. Individual networked micro-controllers can represent up to 20% of the total cost of a typical vehicle in the early 2000s [77], and automotive electronics are present at each level in a modern vehicle [50]. The vehicles are not the only part of transport ecosystem that have become smart and connected. The road network also becomes smarter with intelligent traffic lights and traffic sensors on roads.

In the 2000s, these intelligent systems interact in local networks thanks to the evolution of the communication networks, the information can be shared with all the users of the road networks. This increased ability allows for more sophisticated applications to be implemented, relying on model-based control and artificial intelligence. Intelligent Transportation Systems will enable various users to be better informed and make safer, more coordinated [28, 21], and "smarter" use of transportation networks.

1.1.4 Thesis positioning

This thesis has for purpose to tackle these environmental issues by combining these two new technological solutions, hybridization and Intelligent Transport System. The main goal is to find a way to use the new information available thanks to the ITS to improve the consumption of the hybrid electrical vehicle.

1.2 Hybrid Electrical Vehicles' Architecture

In the end of the 20^{th} century, different attempts have been done to design hybrid vehicles. And in the beginning of the 21^{st} century, the hybrid electric vehicle has conquered its own place in the automotive market, essentially due to the rise of the oil price and binding emission targets. Car manufacturers have developed different ways to use the electric energy in addition to the internal combustion engine. The main principles and concepts used by the HEVs are presented below.

1.2.1 Hybrid Configurations

Usually, the HEVs are classified in four main types according to [15]:

- **Series Hybrid**: The electric engine alone drives the vehicle. It can be supplied by the IC engine or battery.
- **Parallel Hybrid**: The IC engine and the electric one drive the vehicle together simultaneously or separately.
- **Combined Hybrid**: Such vehicles have the ability to operate as series or parallel hybridization.
- **Complex Hybrid**: As reflected by its name, this system involves a complex configuration which cannot be classified into the above three kinds. It will not be described here.

Series Hybrid

In a series HEVs, the IC engine works as an auxiliary power unit and thus extend the range of a purely electric vehicle, see Fig.1.3. This configuration has the advantage that the IC engine is not related to the mechanical requirement of the vehicle and so can be used at a point where efficiency and emissions are optimal. Furthermore the absence of mechanical link between the vehicle and the IC engine decreases the loss due to the gears or clutch. The regenerative braking is still possible by using the electric motor as

a generator. However, such configuration requires one IC engine, one electric generator and one electric motor. Such additional weight can cancel the previous benefits.

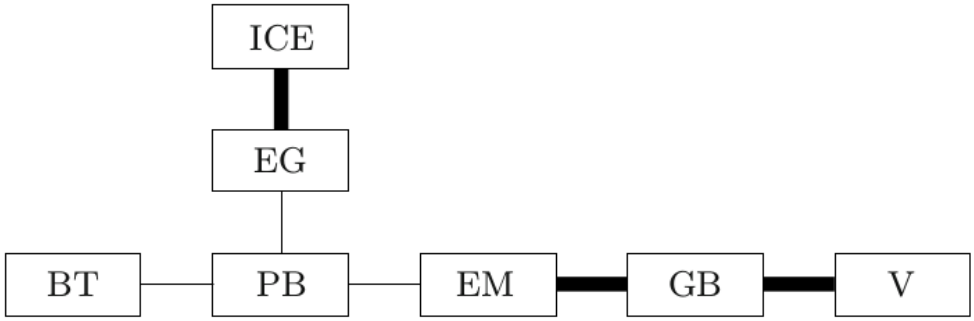


Figure 1.3 – Series hybrid configuration where ICE = IC engine, EG = electric generator, BT = battery, PB = Power link, EM = electric motor, GB = gear box, V = vehicle. Bold lines means mechanical link, solid lines means electric link.

Parallel Hybrid

In a parallel HEV, the engine IC works at the same time as the electric motor. This allows the vehicle to choose the part of each energy used to propel the car, see Fig.1.4. The electric motor can propel the car alone or in combination with the IC engine, thanks to that the working point of the IC engine can be adjusted to optimize the efficiency and emissions. Using the electric motor as an electric generator, we can use regenerative breaking, and thanks to the torque coupler it is possible to use the IC engine to charge the battery. The main disadvantage is the presence of mechanical links which create losses of energy because of the frictions.

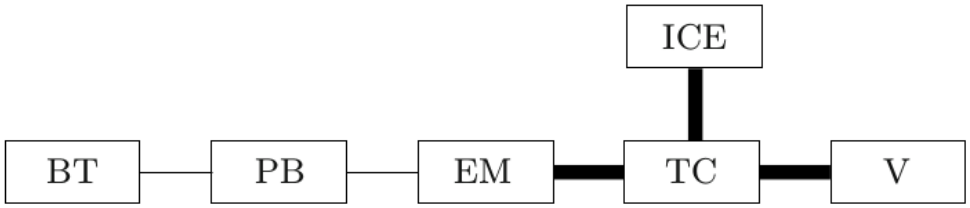


Figure 1.4 – Parallel hybrid configuration. Same nomenclature as Fig. 1.3. (TC = torque coupler)

Combined Hybrid

The combined HEV is a combination between the series type and the parallel type, see Fig.1.5. A HEV of this type contains, as a series HEV, an electric generator and an electric motor. But, as for a parallel HEV, there is also a torque coupler between

the electric generator, the electric motor and the internal combustion engine. Even if combined vehicles have the disadvantages of the two previous combinations, they acquire flexibility by splitting the power between the two motors and because the power provided by the IC engine is decoupled, it can be used at optimal operating point.

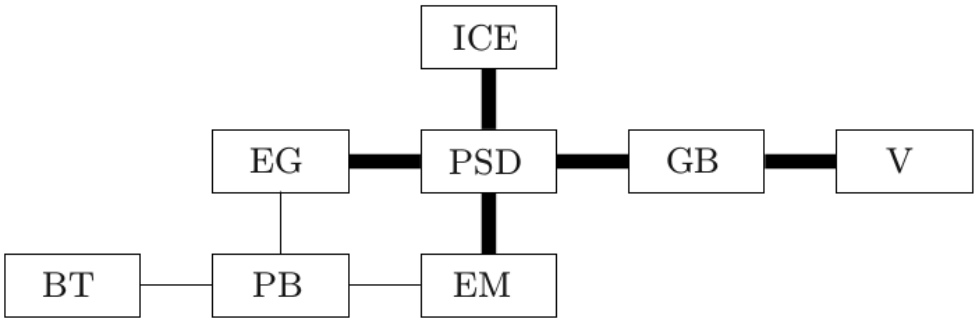


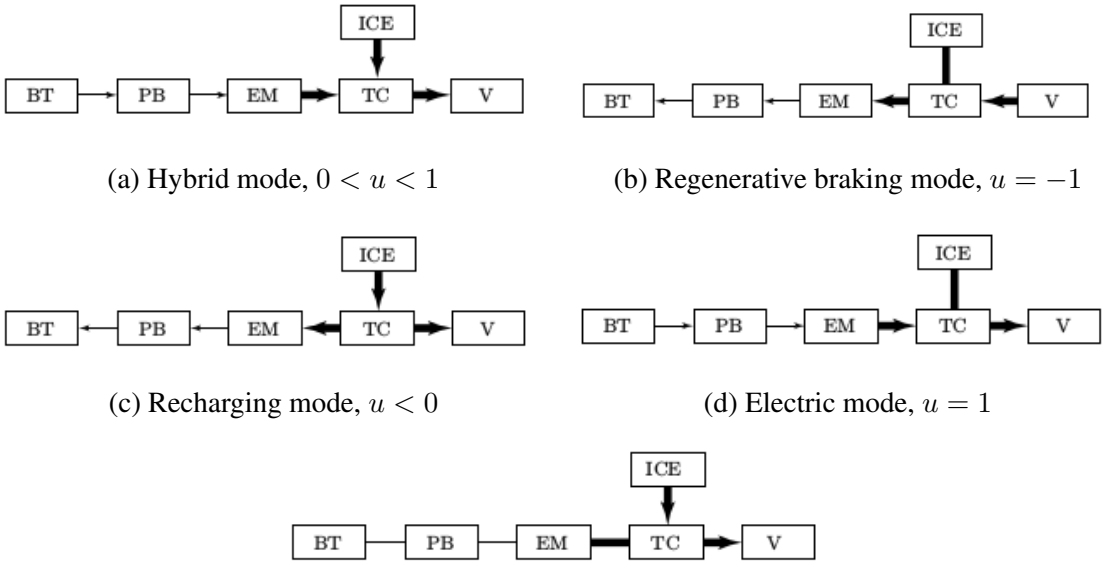
Figure 1.5 – Combined hybrid configuration. Same nomenclature as Fig. 1.3 and Fig. 1.4. (PSD = Power Split Device)

These three types of HEV work in a same way and they have the same possibilities, the main difference remains at the point where the combination between the power delivered by the ICE engine and the power delivered by the battery is done. For sake of simplification, in the following parts we will take the example of the parallel hybrid to explain our claims.

1.2.2 Operating modes of HEVs

A hybrid car is propelled by two engines, using two different sources of power. In this thesis, the car has a fuel engine and an electric one. Hybrid vehicles are complex due to the dispatch of the load between the two engines. This dispatch creates various operating modes with respect to the proportion of load attributed to each engine. We denote u the proportion of energy from/to the electric motor over the total power at the torque coupler. There are five operating modes depending of the value of u:

- **Hybrid mode**: the ICE provides only a portion of the power needed, the electric motor provides the rest, Fig.1.6a ($0 < u < 1$).
- **Regenerative braking mode**: the speed of the vehicle decreases, and the electric motor acts as a generator, recharging the battery, Fig.1.6b ($u = -1$).
- **Recharging mode**: here the engine provides more power than needed to move the vehicle, the electric motor once again acts as generator, Fig.1.6c ($u < 0$).
- **Electric mode**: the motor provides all the power itself, Fig.1.6d ($u = 1$). The internal combustion engine is turned off, and no fuel is consumed.
- **ICE mode**: the engine provides all the power itself, Fig.1.6e ($u = 0$). The electric motor is turned off, and the battery state of charge remains constant.



(a) Hybrid mode, $0 < u < 1$

(b) Regenerative braking mode, $u = -1$

(c) Recharging mode, $u < 0$

(d) Electric mode, $u = 1$

(e) ICE mode, $u = 0$

Figure 1.6 – Operating modes

1.3 Energy Management Systems for Hybrid Electric Vehicle

Using two engines gives the possibility to control the origin of the power. Either electric power or combustion power. Thanks to this control, it is possible to optimize the consumption of the Hybrid vehicle. The energy management system (EMS) determines the split between the internal combustion engine (ICE) and the electric motor.In our case we are interested in improving the fuel economy. Several energy management systems have been developed for HEV, see [63] for a detailed review. Fig.1.7 shows the related classification of the strategies based on the adopted approach. In the following section, the main approach families are presented.

1.3.1 Rule-based EMS

The first class of energy management systems is the one based on rules, determined thanks to heuristic and mathematical models. They use preset rules to determine the power balance at real time, using various information from the vehicle's sensors. The decision rules can either be deterministic (binary logic), or use fuzzy logic (notion of partial truth). These fuzzy logic based energy management have shown good results since the early 00's [80], and are now even able to determine gear shifting [86]. The main disadvantages of these methods are that they are based only on the instantaneous outputs and that their results are not guaranteed to be optimal. Indeed, theses rules are determined by the car manufacturer based on standard speed profile that are not always representative of the reality.

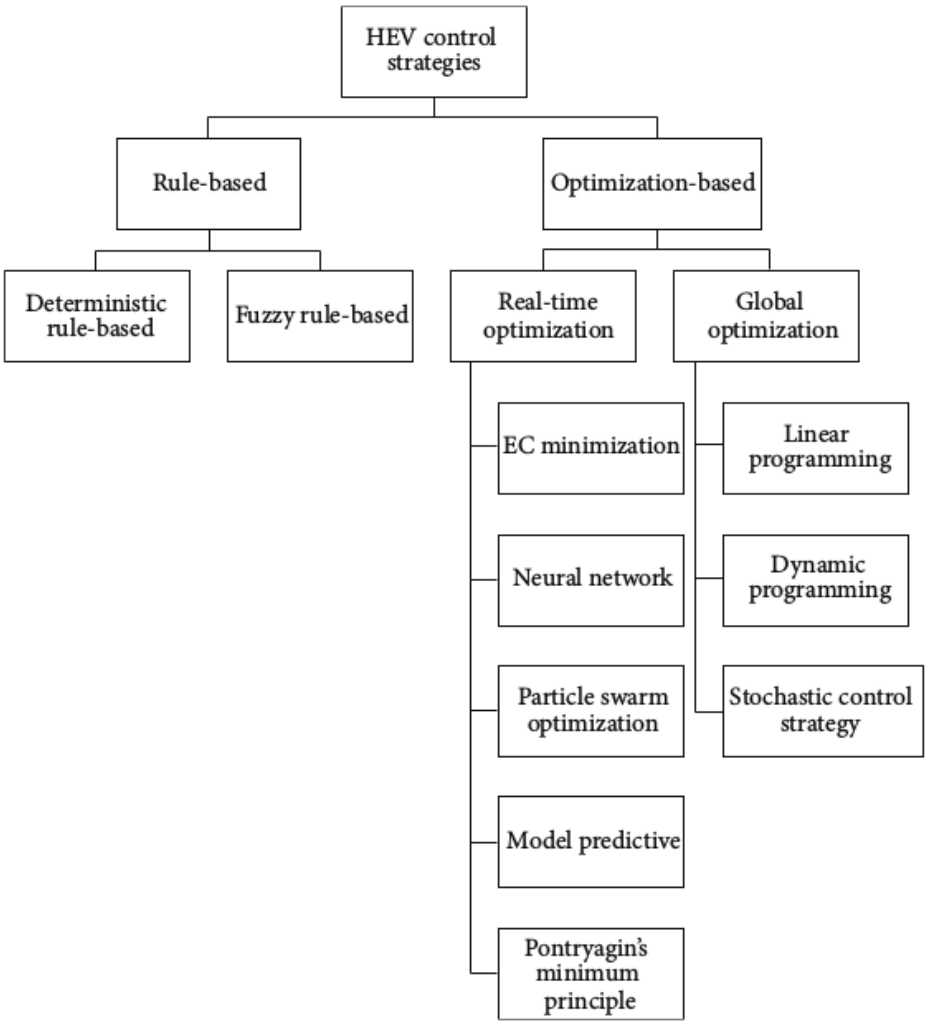


Figure 1.7 – Classification of the control strategies, source: [73]

1.3.2 Optimization-based EMS

The optimization-based strategies are based on optimization theory, more precisely optimal control theory. Their goal is to minimize an objective function, usually in our framework consumption or emissions. They are split in two main classes, the ones trying to compute in real time a local solution, and the others computing a global solution, usually off-line.

Real-time optimization

The real-time strategies aim to obtain a split of the power between the ICE and the motor which is feasible at each time and also to minimize the cost function. This class of strategy has been widely developed during the last years due to relatively easy implementation. The main strategies developed are:

- **Equivalent Consumption Minimization (ECM)**:
  In 2002, Panagelli et al. [72] proposed the concept of equivalent fuel consumption for energy management system. The ECM is developed by calculating the total fuel

consumption as the fuel used by the ICE and also an equivalent fuel consumption for the electric motor. Thanks to this method, the electric power and the combustion power are unified. Adaptive method [69] based on this ECM has been developed, that refreshes the equivalent factor while taking into account driving conditions. The equivalent consumption is computed in real-time, as a function of the current output of the system. The method allows to work without any information over the future driving conditions. In [75], a comparison between ECMS and rule-based strategies is proposed, and shows that ECMS strategies outperform.

- **Model Predictive Control (MPC)**:
  MPC is based on an online system parameters update combined with optimal control tools. This allows to predict the future speeds and therefore optimize the consumption by taking them in account. The optimization is done over a finite time horizon and the solution is then implemented in the current time slot only. The method is repeated for each time slot. An extended review of MPC-based energy management systems is proposed in [45].

- **Pontryagin's maximum principle (PMP)**: In a deterministic framework, so-called indirect method, based on PMP, can also be implemented in order to find the optimal control, see [52, 51, 91].

Global optimization

The strategies developed to find a global optimal solution are essentially based on the concept of Dynamic Programming (DP). This method solves the problem by breaking it into easier subproblems, thanks to the Bellman's optimality principle.

Deterministic Dynamic Programming (DDP) has been widely used in order to obtain theoretical lower bound for the consumption on specific speed profile, and a rule-based EMS relying on the obtained optimal policy obtained has been proposed in [60]. The main drawback of such methods is the curse of dimensionality, i.e. the computational cost typically increases exponentially with the number of states (and control) variables. Then this method seems limited to small systems and can be very difficult to use in real time, although some works propose dimension reduction techniques for the state space or alternatly on the control space, see [76].

In order to manage uncertainty of the future speeds, stochastic formulations have been proposed and solved thanks to stochastic DP (SDP) [14, 87]. Unfortunately, this further increases the computational cost of the method.

1.4 Thesis contributions

The main results of the thesis are that an optimal control taking into account the stochastic aspect of traffic can be obtain off-line, and the optimal control reused in order to reach a global optimum in real-time.

The contributions of the thesis are twofold:

- **A stochastic traffic model**, based on clustered (speed/acceleration) probability distributions, able to efficiently estimate the energy consumption distribution of vehicles on a road segment, with a reduced data occupancy.

- **A bi-level EMS** taking in account the stochastic traffic conditions, to compute online the optimal state of charge and torque trajectories.

In the following, the contributions of each chapter are presented.

Chapter 2 - A stochastic data-based traffic model

A new approach to estimate traffic energy consumption via (speed,acceleration) probability distributions is proposed. For each segment s, we record the (speed,acceleration) of all the vehicles passing through the segment during all the time-frames $t_{i,s}$. Thanks to these measures, empirical (speed,acceleration) probability distributions can be obtained on each segment composing the road network.

In order to reduce data occupancy, clustering techniques are used to obtain meaningful classes of traffic conditions. The distance used in the clustering algorithm is the 1-Wasserstein distance, based on optimal transport:

$$W_1(\mu,\nu) = \min_{\pi\in\Pi(\mu,\nu)} d\cdot\pi \tag{1.4.0.1}$$

where $\Pi(\mu,\nu)$ is the set of transportation plans from μ to ν, i.e. the set of nonnegative matrices π of size $(|N_{\mathbb{V}}|\times|N_{\mathbb{A}}|)^2$ with marginals $\pi^\top\mathbf{1}=\mu$ and $\pi\mathbf{1}=\nu$. The barycenters, obtained thanks to this distance, keep the geometrical aspects of the initial probability distributions. Different times of the day with similar speed patterns and traffic behaviour are thus grouped together in a single cluster, Fig.1.8.

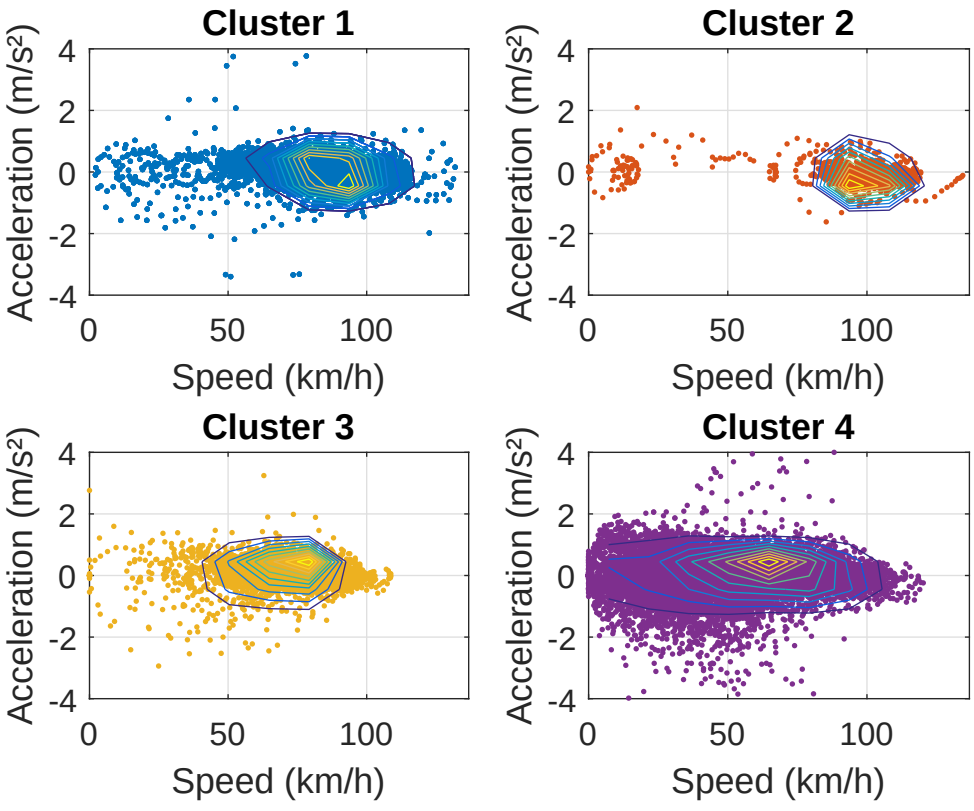


Figure 1.8 – Illustrative example of clusters with their initial aggregated data.

Different energy consumption models based on the aggregated data are proposed to estimate the energy consumption of the vehicles in the road network.

The first model consider the expectation of the power $P(v,a)$ according to the probabilities induced by the law of the barycenter $\bar{\mu}$ of the traffic cluster :

$$\bar{P}(\bar{\mu}) = \sum_{\mathbb{V}\times\mathbb{A}} \bar{\mu}_t(v,a)P(v,a) \tag{1.4.0.2}$$

This average power is integrated over the time interval $[t_i, t_f]$ spent by the vehicle on the segment, thus

$$C_{Avg} = \int_{t_i}^{t_f} \bar{P}(\bar{\mu}_t)dt \tag{1.4.0.3}$$

The second method is based on the idea that the vehicle follows the traffic, in a statistical sense. Therefore the method generates a sequence of (speed,acceleration) based on the traffic distribution $\bar{\mu}$ that is integrated in order to obtain the consumption.

For validation purposes, a traffic simulator is used to generate the data and compare the estimated energy consumption to the measured one. A thorough sensitivity analysis with respect to the parameters of the proposed method (i.e. number of clusters, size of the distributions support, etc.) is also conducted in simulation. Finally, a real-life scenario using floating car data is analysed to evaluate the applicability and the robustness of the proposed method.

The results show that clustering techniques allow to reduce data occupancy while retaining most of the useful information in order to estimate the energy consumption distribution.

Chapter 3 - An Optimal local policy using traffic prediction

A method to obtain an optimal policy that takes into account the traffic behaviour is presented.

Our basic assumption is that the speed and acceleration of the vehicle $(\mathbf{V}, \mathbf{A})$ are a piecewise i.i.d. process following the successive traffic distributions. Such distributions have been obtained in chapter 2. Therefore, the traffic behaviour is taken into account to obtain an optimal motor torque T_m adapted to the load. This control minimizes the fuel consumption C of the engine according to a target final state of charge SoC_f and the power map P_m of the motor. Finally a discrete stochastic optimal problem can be written on a segment of length L_s :

$$\min_{T_m} \quad \mathbb{E}\left[\sum_{k=0}^{k_f} h^k C(\mathbf{V}^k, \mathbf{A}^k, T_m^k,) + P(SoC_f, SoC^{k_f})\right] \tag{1.4.0.4}$$

$$s.c\ \forall k, \quad SoC^{k+1} = SoC^k + \frac{h^k}{C_{max}} P_m(\mathbf{V}^k, \mathbf{A}^k, T_m^k) \tag{1.4.0.5}$$

$$D^{k+1} = D^k + h^k \mathbf{V}^k \tag{1.4.0.6}$$

$$T_m^k \in [T_{min}, T_{max}] \tag{1.4.0.7}$$

$$SoC^k \in [0, 1] \tag{1.4.0.8}$$

$$h^k := \min(h_0, (L_s - D^k)/\mathbf{V}^k) \tag{1.4.0.9}$$

$$k_f := 1 + \max\{k, D^k + h_0 \mathbf{V}^k < L_s\} \tag{1.4.0.10}$$

$$SoC^0 = SoC_0;\ D^0 = 0. \tag{1.4.0.11}$$

Solving this problem for all possible initial states of charge, thanks to stochastic dynamic programming, will give us an expected consumption according to the final state of charge and the optimal policy in order to realize it, see Fig.1.9.

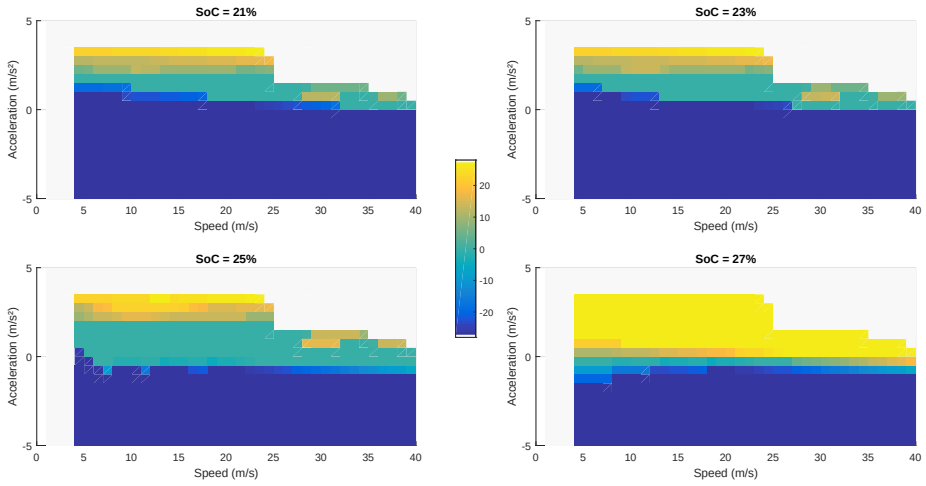


Figure 1.9 – Optimal motor torque $T_m(v, a)$ obtained for a fluid traffic.

The error due to the discretization as well as the computation have been studied.

Numerical comparisons on real data collected on the A7 french highway with the deterministic optimization have been done. The deterministic optimization gives the minimum consumption reachable for a vehicle knowing its speed profile. The results indicate that the stochastic method proposed in this chapter handles the traffic conditions with a reasonable overconsumption, almost 5% of the deterministic consumption.

Chapter 4 - An Optimal State of charge trajectory on a travel with predicted traffic

Due to the computational cost of the stochastic dynamic programming, it is possible to solve it on small road segments, but increased lengths and therefore travel times prevent real-time implementation. In order to obtain on-line optimal policies adapted to the travel, a bi-level method, based on segment decomposition, is proposed.

The method of chapter 3 computes the optimal motor torque on a road segment s knowing the traffic conditions, represented by a probability distribution μ_s. The stochastic dynamic programming can be done off-line on predefined traffic conditions, the clusters computed in chapter 2. By solving these problems off-line, cost maps $\mathcal{C}$ depending on the initial and the final state of charge are defined and stored, see Fig 1.10.

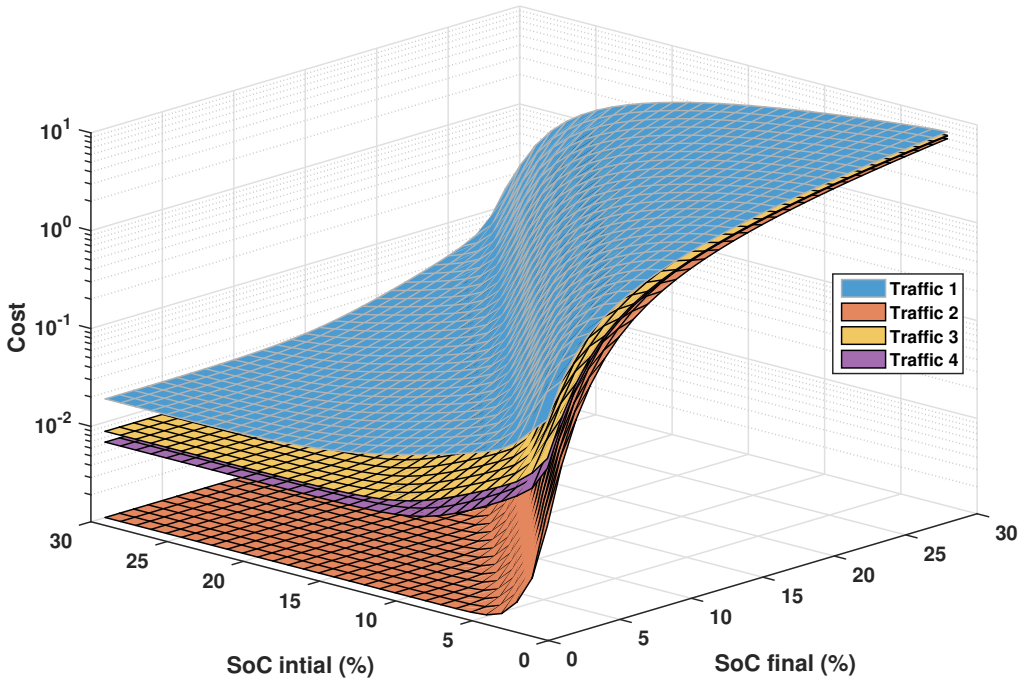


Figure 1.10 – Illustrative cost maps

These cost maps are used to predict the mean cost of segment according to the state of charge at the beginning and the end of the segment. Therefore, a new problem that computes the reference state of charge trajectory SoC^r to be reached at the end of each segment, based on these cost maps, write as follows:

$$\min_{SoC^r} \quad \sum_{s=0}^{S-1} \mathcal{C}(SoC_s^r, SoC_{s+1}^r, \mu_s) \tag{1.4.0.12}$$

$$s.t.\ \forall s, \quad SoC_s^r \in [0,1] \tag{1.4.0.13}$$

$$SoC_0^r = SoC_{intial} \tag{1.4.0.14}$$

$$SoC_S^r = SoC_{final} \tag{1.4.0.15}$$

The main advantage of this bi-level method is that the uncertainty of the traffic behaviour has been already treated thanks to stochastic based optimizations. Thus the new problem can be solved thanks to a Deterministic Dynamic Programming, and therefore can be computed on-line. The state of charge trajectory is obtained along the whole journey as well as the optimal policy on each segment.

Numerical simulations have been done in order to compare the consumption obtained with a theoretical lower bound, computed thanks to deterministic dynamic programming on speed profiles. The results show that the bi-level method induces an overconsumption ($\sim 15\%$), and satisfy well the final state of charge constraint, while maintaining a simulation time smaller than 10 seconds.

Chapter 5 - Application to the optimal routing problem under traffic conditions

On road networks, multiple paths are possible to arrive to the wanted destination. Traffic conditions on these paths can vary significantly due for instance to speed limitations or traffic jams. In order to minimize the global consumption, we propose to use the methods developed in the previous chapters to find the optimal consumption path, according to consumption and traffic conditions, in the road network.

This one is modelled as a graph, where each road intersections correspond to a node. This graph is extended in each node with the possible state of charge, in order to compute both the optimal path and the optimal state of charge trajectory. The set of all possible paths between the origin S and the destination D is denoted Γ_{SD}. The shortest path problem is read as follows:

$$\begin{aligned} &\underset{\gamma\in\Gamma_{SD}}{\text{minimize}} && \sum_{i\in\gamma} \nu_{N_iN_{i+1}}(SoC^r_{i+1}, SoC^r_i) && (1.4.0.16)\\ &\text{s.t} && \forall i \in \gamma, SoC^r_i \in [0,1] && (1.4.0.17)\\ &&& \forall i \in \gamma, SoC^r_{i+1} - SoC^r_i \in [-a_i, b_i] && (1.4.0.18) \end{aligned}$$

We have established that the discretization error (for the value function) is of the order of the discretization step. Classical A^* algorithm is performed to solve it.

Numerical simulations have shown that the consumption of such eco-path is reduced compared to the fastest one and the SoC_f constraints are relatively satisfied.

Appendix A - A Stochastic gradient method for Wasserstein barycenters of distributions

In this appendix, a new algorithm is presented, based on stochastic gradient methods to compute Wassertein barycenter of probabilities distribution. The fast computation of such barycenter is useful for instance to perform clustering. The Wasserstein distance, based on the optimal transport, compare two distributions. Denoting μ the barycenter of the distribution ν^k and c the associated cost matrix, the transportation plan π is computed as follows:

$$\begin{aligned} &\min_{\pi,\mu} \textstyle\sum_{k=1}^K c\cdot\pi^k; \;\; \pi \geq 0;\\ &(\pi^k)^\top \mathbf{1} = \mu, \;\; \pi^k \mathbf{1} = \nu^k, \;\; k = 1,\dots,K, \end{aligned} \qquad (1.4.0.19)$$

Fast methods to compute these barycenters are required when handling large amounts of data. Thanks to a double entropic regularization of the barycenter problem and using the semi dual form, with u the Lagrange multiplier, of the regularized problem with reverse entropy, we obtain a problem that can be solved thanks to an algorithm of the stochastic gradient class:

$$\max_u \sum_j h^j(u) \qquad (1.4.0.20)$$

with :

$$h^j(u) = -\varepsilon_1 \sum_k \nu^k_j (\log(\frac{1}{\nu^k_j} \sum_{i\in X^0} e^{\frac{-c_{ij}}{\varepsilon_1}} e^{\frac{-u^k_i}{\varepsilon_1}}) + 1) - \frac{\varepsilon_2 \sum_k \nu^k_j}{K} (\sum_{i\in X^0} e^{\frac{\sum_{r=1}^K u^r_i}{\varepsilon_2}}) \qquad (1.4.0.21)$$

The time and spatial complexity of the stochastic gradient has been compared to the Sinkhorn iterations for barycenters introduced in 2015 [8].

CHAPTER 2

A stochastic data-based traffic model

This chapter has been published in IEEE Transactions on Intelligent Transportation Systems in 2019 [56].

Contents

2.1 Introduction

In 2015, according to data from the European Environment Agency, road transportation contributed to 21% of total EU-28 greenhouse gas emissions. In order to meet the long-term emissions reduction target, emissions from transportation need to fall by more than two thirds by 2050 [33]. These emissions are essentially a function of the vehicle propulsion technology and the driving style [34].

Estimating energy consumption of the vehicles is a great challenge in the objective of improving global transportation efficiency, since this information is used in energy management, eco-routing, eco-driving, traffic management, ... Traffic congestion has a major impact on the driving behavior, and thus plays a key role in the level of fuel consumption [90].

Therefore, accurate predictions of vehicles energy consumption must take traffic conditions into account. To perform this objective, faithful modeling of traffic behavior is of primary importance. Energy-oriented modeling approaches can be divided in two main categories.

On the one hand, several mathematical traffic models are available nowadays, see for instance [88]. Such models typically depict the reality either from a macroscopic point of view, based on the road vehicular density [59], or from a microscopic perspective, based on the description of the instantaneous behavior of each vehicle [74]. Both approaches have limitations in providing an accurate energy consumption estimation. Macroscopic models typically provide average traffic speeds to compute energy consumption [94], thus neglecting the impact of speed fluctuations due to congestion. Higher precision of the energy consumption estimation could only be obtained at the expense of a denser discretization of the road network, therefore compromising scalability. Microscopic models could achieve precise energy consumption estimation, but they require a significant calibration and validation effort. Also, the computational burden and the amount of collected data grows rapidly with the size of the network, therefore these models are more suitable for off-line use.

On the other hand, data-based models rely on collected traffic information to estimate traffic behavior and energy consumption. Instantaneous models are able to precisely estimate energy consumption by using large amounts of data, generally the measured driving profile of each vehicle. To tackle this drawback, aggregated models use the average value of the measured speed profiles to compute energy consumption, but they suffer from the same accuracy problems previously discussed for the macroscopic traffic models. Furthermore, the data sparsity and availability is an issue [42]. Other approaches try to solve the problem of the data sparsity by simply classifying road segments by category (e.g. urban, arterial, freeway, etc.), in order to associate each category with a typical energy use. This type of models may lead to inaccuracy in energy consumption estimation, as road segments belonging to the same category may show very different traffic patterns [31].

In this work, a new way to represent traffic behavior on large road networks is proposed. The objective of this model is to accurately depict the effect of traffic conditions on the vehicles energy consumption in each road segment. The key idea is to use a statistical approach based on vehicle speed and acceleration data, measured from real vehicles. In particular, the entire observation time during which speed and acceleration data are collected is subdivided into time-frames. During each time-frame a (v, a) distribution is

generated, and such a distribution is then used as an input for an energy consumption model to estimate the traffic energy consumption on the analyzed road segment during the specific time-frame. Therefore, each road segment is defined by its own collection of (v, a) distributions.

The proposed model also includes the possibility of reducing the dimension of the traffic data and increasing scalability, by applying clustering techniques to the probability distributions of each road segment. For instance, the different distributions representing traffic in one road segment over different hours of the day may be aggregated in clusters modeling only significant traffic conditions (e.g. peak, off-peak, etc.).

The chapter is organized as follows. The clustering technique and the proposed energy consumption model are presented in Section 2.2. The traffic data collection and the model validation procedure are discussed in Section 2.3. Using traffic data from the simulator SUMO, Section 2.4 illustrates the method first on a single road segment, then for a larger set of segments. Finally, section 2.5 presents an application of the method to actual traffic data, as well as a comparison to a more basic approach, using only mean speeds.

2.2 Proposed Method

2.2.1 Road segments

In the following, the road network is assumed to be subdivided into a collection of *segments*. A segment is a portion of road with homogeneous topographic characteristics. Segments are typically delimited by network elements such as traffic lights, crossroads, or roundabouts. With a traffic simulator such as SUMO we can retrieve the segments from the simulator model, see 2.4.1. When working on actual traffic data, segments can be obtained from Geographic Information Systems (GIS) data (e.g. OpenStreetMap, HERE Maps, etc.). Segments length typically range between a few meters and a few hundreds meters. In the following we denote S the set of road segments, of size N_S.

2.2.2 Probability distribution for (speed,acceleration)

The consumption of an engine depends on its operating point, which can be determined by the speed and acceleration of the vehicle. The key point in our method is to assume that an accurate (with respect to energy consumption) description of the traffic can be derived from the probability distributions of measured speeds v and accelerations a for each road segment. Note that these distributions do not retain the temporality of the speed profiles.
We set N_T the number of time-frames for each segment[1]. We denote the family of time-frames $(t_{i,s})_{(1...N_t)\times(1...N_S)}$, and their length $(\Delta T_{i,s})_{(1...N_t)\times(1...N_S)}$. For each segment s, we record the (speed,acceleration) of all the vehicles passing through the segment during all the time-frames $t_{i,s}$.

In the following, we work with discrete distributions in the (v, a) space. We denote $\mathbb{V}$ and $\mathbb{A}$ the sets of feasible speed and acceleration. To simplify, these sets are taken identical for all segments and time-frames, thus all (speed,acceleration) discrete probability

[1]For the sake of simplicity, we assume the same number of time-frames for all the segments

distributions have the same support in $\mathbb{V} \times \mathbb{A}$. We denote $N_{\mathbb{V}}$ and $N_{\mathbb{A}}$ the discretization size of $\mathbb{V}$ and $\mathbb{A}$. Recalling that N_S and N_T are the number of road segments and time-frames, we obtain a total of $N_S N_T$ discrete distributions of support size $N_{\mathbb{V}} N_{\mathbb{A}}$. Fig.5.2 shows an example of such a distribution.

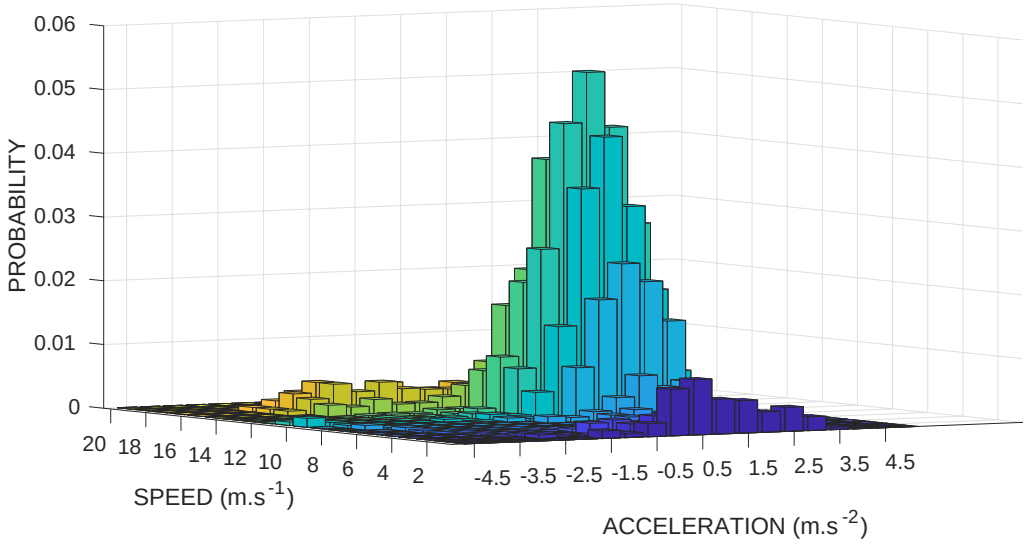


Figure 2.1 – Example of (speed,acceleration) distribution.

2.2.3 K-means clustering with strong patterns

For each segment s and time-frame $t_{i,s}$ we have a probability distribution $\mu_{i,s}$. When considering large road networks with several thousands of segments, the data size grows rapidly to several gigabytes. A natural idea is to reduce the information through clustering techniques applied to these distributions. Then, for each cluster j, we can take its barycenter $\bar{\mu_j}$ as representative of the traffic conditions for all the segment/time-frame pairs $(s, t_{i,s})$ that belong to this cluster. This way, we only have to store the distributions corresponding to the barycenters of the clusters.

Since the elements to be clustered are probability distributions, we use the 1-Wasserstein distance ([89, 26]). This distance based on optimal transport theory tends to preserve the geometrical aspects (shapes) of the distributions. We recall the definition of the Wasserstein distance:

$$W_1(\mu, \nu) = \min_{\pi \in \Pi(\mu,\nu)} d \cdot \pi \tag{2.2.3.1}$$

where $\Pi(\mu, \nu)$ is the set of transportation plans from μ to ν, i.e. the set of nonnegative matrices π of size $(|N_{\mathbb{V}}| \times |N_{\mathbb{A}}|)^2$ with marginals $\pi^\top \mathbf{1} = \mu$ and $\pi \mathbf{1} = \nu$. As speed and acceleration have comparable magnitude orders when expressed in SI units, we simply define the displacement cost d by the Euclidean norm between the origin and destination points.

The notion of barycenter for distributions has been extended to the Wasserstein distance in [1], where the barycenter is defined as the distribution that minimizes the sum of

the squared Wasserstein distances to all distributions of the set. For the practical aspects we follow [8] in which the barycenter minimizes the sum of W_ϵ distances, with W_ϵ an entropic regularization of the Wasserstein distance, computed by a Sinkhorn-type algorithm. Due to the possible large dimension of the speed and acceleration grid, another algorithm SAG has been explored to handle high dimensional distributions. This algorithm is presented in Annexe A.

We use the K-means algorithm [62] to compute the clusters, and note k the (fixed) number of clusters. To reduce the sensitivity of the K-means algorithm to the initial guess, we use the strong patterns method [58]. Strong patterns are subsets whose elements always end up in the same cluster regardless of the K-means starting point. In practice, we run a first batch of K-means with random initializations to identify the strong patterns, and then perform a final K-means which is initialized by taking one element in each of the k largest strong patterns. The principle of the K-means is recalled in Algorithm 1.

Algorithm 1: K-means algorithm

Input : Distributions to be clustered $\mu_1 \cdots \mu_I$
Output: Clusters barycenters $\bar{\mu_1} \cdots \bar{\mu_k}$

Initialization
for $j \in \{1, \ldots, k\}$ **do**
  $\bar{\mu_j} \leftarrow$ choose randomly in $\mu_1 \cdots \mu_I$

Iterate until convergence
Initialize barycenter backups $\hat{\bar{\mu}}_j \leftarrow 0$
while $\exists j \in \{1, \ldots, k\}, \bar{\mu_j} <> \hat{\bar{\mu}}_j$ **do**
  Backup all barycenters $\hat{\bar{\mu}}_j \leftarrow \bar{\mu_j}, \forall j \in \{1, \ldots, k\}$
  for each distribution μ_i **do**
    Find the closest barycenter $\bar{\mu_c}$
    Label μ_i as part of cluster c
  Update all cluster barycenters $\bar{\mu_j}, \forall \{j \in 1 \ldots k\}$

2.2.4 Computing energy consumption

The energy consumption computed is the energy at the wheel, neglecting the losses due to the powertrain. The instantaneous power at the wheel is denoted as a general function $P(v, a)$, which can be for instance of the form presented in (2.3.1.1). The ultimate objective of the proposed method is to estimate energy consumption by using only the information extracted from the (v, a) probability distributions μ. More accurately, we seek to obtain the energy consumption from the barycenter $\bar{\mu}$ of the cluster containing μ. In the following we introduce two methods to compute the consumption of a generic vehicle passing through a segment. These will be referred to as "Average Consumption Method" and "Memoryless Sampling Method".

Average Consumption method

The first idea is to use the average power $\bar{P}$ (in the probabilistic sense)

$$\bar{P}(\bar{\mu_t}) = \sum_{\mathbb{V}\times\mathbb{A}} \bar{\mu_t}(v,a)P(v,a) \tag{2.2.4.1}$$

where $\bar{\mu_t}$ is the barycenter of the cluster containing the current segment at time t. The barycenter may indeed change if t crosses different time-frames while the vehicle is on the segment. This average power is integrated over the time interval $[t_i, t_f]$ spent by the vehicle on the segment, thus

$$C_{Avg} = \int_{t_i}^{t_f} \bar{P}(\bar{\mu_t})dt \tag{2.2.4.2}$$

Note that knowledge of the time interval is required in this method, in addition to the (speed,acceleration) distributions. Indeed, using here the average time would give identical consumption for every vehicle. Therefore, we need some more statistically significant time information in order to capture the deviation of the consumption distribution. A resulting drawback of this method is that a faster vehicle has a shorter travel time and thus a lower energy consumption, which may seem unrealistic.

Memoryless Sampling method

In the second method the energy consumption is still obtained by integrating the instantaneous power, but we do not use the average power. Instead, we implement the idea that the vehicle must follow the traffic at every time, in a statistical sense. More precisely, its speed and acceleration should follow the probability distribution of the barycenter $\bar{\mu}$ of the cluster for the current pair (s,t). Another difference is that the integration is performed over the segment length L_s instead of travel time.

So the Memoryless Sampling method generates a sequence of (v_n, a_n), independent samples according to the probability distribution $\bar{\mu}$. Setting a time step δt, we use this sequence to integrate both the traveled distance and the instantaneous power. We assume $\delta t = 1s$, in order to have the same order of magnitude as the reaction time of a driver. We stop the generation of (v_n, a_n) when the vehicle reaches the end of the segment [2]. Since the distance will be covered in a finite time, we obtain the finite set of samples $(v_n, a_n)_{n=1,\dots,n_f}$ and the consumption writes as

$$C_{MSM} = \sum_{n=1}^{n_f} P(v_n, a_n)\delta t \tag{2.2.4.3}$$

2.3 Validation Approach

2.3.1 Power and Reference energy consumption

In order to assess the accuracy of our two methods (Average and Memoryless Sampling), both based on the statistical representation of traffic, we introduce a "reference"

[2] which happens with probability 1 since vehicles never stop indefinitely.

energy consumption for the sake of comparison. First we choose a simple equation for the instantaneous power $P(v, a)$, neglecting all road slope effects:

$$P(v, a) = (ma + \alpha_2 v^2 + \alpha_1 v + \alpha_0)v \tag{2.3.1.1}$$

where m is the vehicle mass and $\alpha_0, \alpha_1, \alpha_2$ define a vehicle-dependent polynomial approximation of the road-load force. We use the numerical values $m = 1400kg$, $\alpha_0 = 185.4$, $\alpha_1 = 0$, $\alpha_2 = 0.3$, corresponding to a passenger vehicle [22]. We would like to point out that energy consumption models are a wide topic, and we refer interested readers for instance to [44]. In our case we essentially want to compare the consumptions from raw traffic data and statistically processed data, so the key point is to use the same power expression for all consumption formulas. In all the following, these energy consumptions C are computed by integrating [3] the instantaneous power P along the (speed,acceleration) profiles, without taking into account any regenerative braking, namely

$$C = \int_{t_i}^{t_f} \max(0, P(\mathbf{v}(t), \mathbf{a}(t)))\mathrm{dt} \tag{2.3.1.2}$$

In order to compare the different methods, we compute the distribution of the energy consumptions for a given (segment,timeframe):
- the Reference energy distribution $\mathbf{C}_{Ref}$ is obtained by plugging into (2.3.1.1) the recorded speed and acceleration $(\mathbf{v}(t), \mathbf{a}(t))$ of all vehicles passing through the segment during the timeframe, according to the traffic simulator.
- for the Average and Memoryless Sampling methods, we recall that the consumption distributions $\mathbf{C}_{Avg}$, $\mathbf{C}_{MSM}$ are obtained using the speed/acceleration probability laws $\bar{\mu}$ from the barycenters, see (2.2.4.2) and (2.2.4.3).

2.3.2 Indicators

Since our aim is to compare distributions of energy consumptions, we study several indicators.

- Mean and Standard deviation errors are classical indicators. We compute these relative errors as follows, with $\mathbf{C}_{method}$ denoting either the Average consumption or the Memoryless Sampling consumption:

$$\varepsilon_{mean}(s) = \frac{\overline{\mathbf{C}_{method}(s)} - \overline{\mathbf{C}_{ref}(s)}}{\overline{\mathbf{C}_{ref}(s)}}$$
$$\varepsilon_{\sigma}(s) = \frac{\sigma_{\mathbf{C}_{method}}(s) - \sigma_{\mathbf{C}_{ref}}(s)}{\sigma_{\mathbf{C}_{ref}}(s)}$$

- Kullback-Leibler divergence [54], also called 'relative entropy', is a particular case of φ-divergence. KL divergence can be used to measure distances between two probability distributions P and Q, however it is not a metric (no triangular inequality or symmetry). Another drawback is that it cannot be computed for instance when the probability of the model q is 0 while the probability of the reference p is not.

$$KL(P|Q) = \sum_i p_i \log(\frac{p_i}{q_i}) \tag{2.3.2.1}$$

[3] in practice integration is done by the Euler scheme.

Instead, we use the Jensen-Shannon divergence, which is a symmetrized version of KL divergence, sometimes referred to as 'total divergence to the average' [20]. Note that the square root of the JS divergence is a metric called JS distance [30, 71]. Fig.2.2 illustrates the JS divergence on a Gaussian with noised parameters.

$$JS(P|Q) = \frac{1}{2}KL(P|M) + \frac{1}{2}KL(Q|M) \tag{2.3.2.2}$$

with

$$M = \frac{1}{2}P + \frac{1}{2}Q \tag{2.3.2.3}$$

Here, we have $P = \mathbf{C}_{method}$ and $Q = \mathbf{C}_{Ref}$.

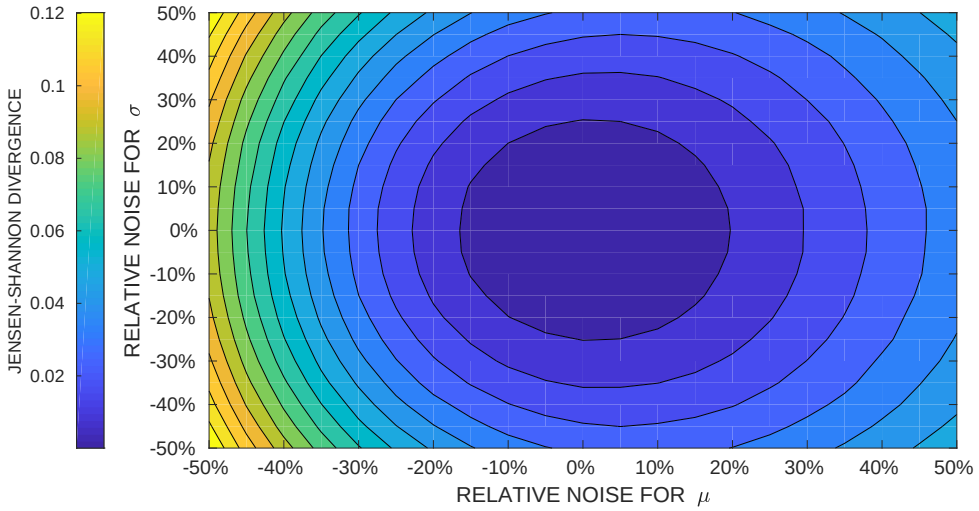


Figure 2.2 – Jensen-Shannon divergence for a noised Gaussian distribution with base parameters $\mu = \sigma = 1$.

2.4 Results with simulated traffic data

2.4.1 Traffic Data from simulation

We illustrate our approach with data obtained from the traffic simulator SUMO [53]. The simulation runs the scenario LuST [16], which models a 24h traffic in the city of Luxembourg. The time step is set to one second, i.e. 1 Hz sampling frequency for the variables of the vehicles. For the road segments we take the subdivision from the scenario, consisting of roughly 24000 elements. We aggregate contiguous road lanes together to obtain more data per segment, and end up with 18322 segments. The length of these road segments is shown on Fig. 2.3.

From the simulation raw data (5.7GB), we extract the segment, speed, and acceleration of each vehicle in the network at all time steps. We aggregate the records for a fixed time-frame length $\Delta t_{s,i}$ (in practice we use a constant frame length Δt for all segments). This is done both to gather sufficient data on the segments, and to decrease the number of traffic distributions for the clustering phase. For instance a time-frame of 1h will give

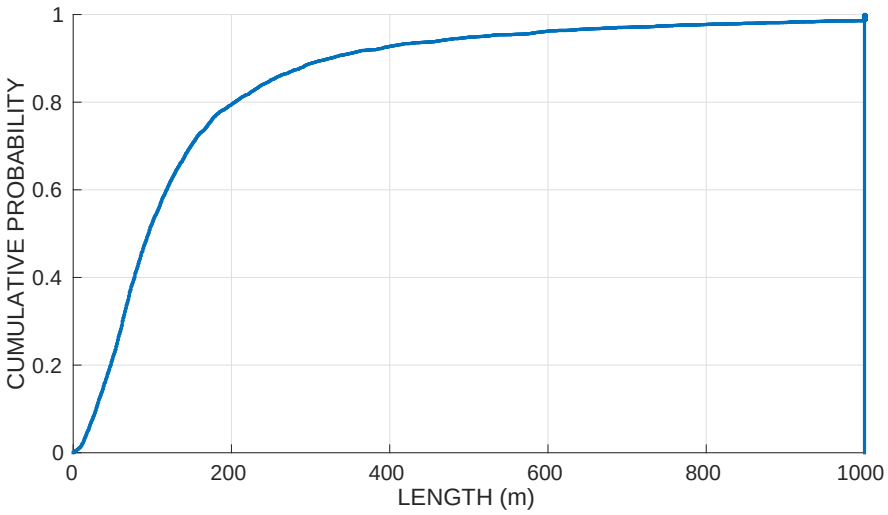


Figure 2.3 – An illustration of cumulative distribution of the length of road segments in city of Luxembourg as implemented in LuST for SUMO.

24 distributions per segment, while 10 min gives 144 distributions per segment. In the following all codes are written in Python/NumPy and run on a standard desktop computer.

2.4.2 Numerical results on one segment

To begin with, we compare the different consumptions on a single road segment. The objects compared are therefore the consumptions distributions of all the vehicles that went through the segment during each time-frame. Results are shown as the cumulative distribution function of the consumptions for all time-frames.

We analyze in particular the influence of the (v, a) discretization, the clustering, and the choice of time-frame duration. Unless specified otherwise, the distributions are shown for a discretization $N_V = N_A = 10$, a time-frame $\Delta t = 10min$, taking the full set of distributions without clustering.

Influence of the (speed,acceleration) discretization

First, Fig.2.4 shows the consumption distribution for both the Average method and Memoryless Sampling method. We test the discretizations $N_V = N_A = 10, 20$ and 30 and compare to the reference consumption. We see that for the Memoryless Sampling method: i) the general shape of the distribution is similar to the reference and ii) finer (v, a) discretizations give distributions closer to the reference. On the other hand, for the Average method: i) we observe some linearization of the consumption and ii) the effect of discretization is much less significant.

Influence of the time-frame

Next we study the effect of the length of the time-frame Δt, which is the time interval over which we aggregate the vehicles data. Longer time-frames may cause some over-

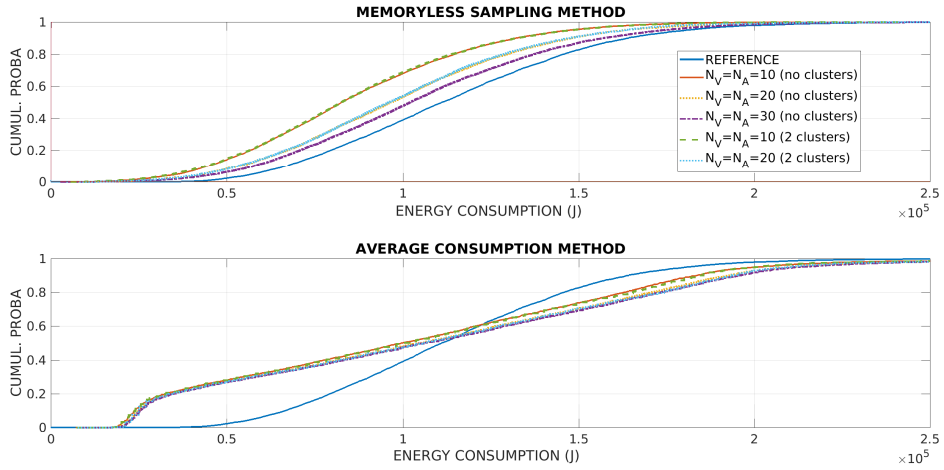


Figure 2.4 – Cumulated consumption distribution for one segment - effect of (v, a) discretization and clustering. Reference consumption (blue line) is recomputed along each vehicle speed/acceleration profile. The two graphs show the consumptions estimated by the memoryless sample and the average consumption methods. We compare several settings for the speed/acceleration discretization, as well as the number of clusters.

averaging and loss of specific traffic information. On the other hand, shorter time-frames may lead to insufficient vehicle data (for statistical relevance), and also increase the number of (v, a) distributions to handle. Fig. 2.5 shows the energy consumptions obtained for time-frames of $5s$, $1min$ and $10min$. On this segment, for both the Average and the Memoryless Sampling methods, the influence of the time-frame duration seems rather small.

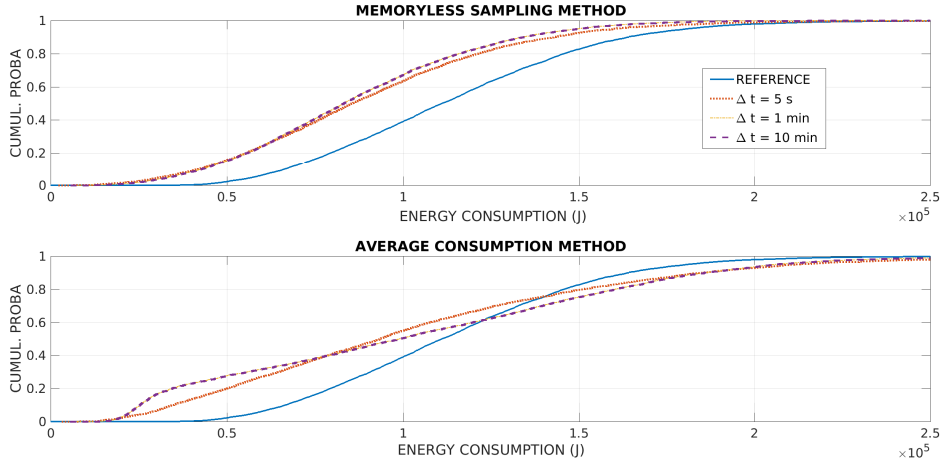


Figure 2.5 – Cumulated consumption distribution for one segment - effect of the time-frame duration.

Influence of the clustering

To conclude this first batch of results, we examine the information loss due to the clustering stage. The consumptions obtained using only the barycenters (green and cyan curves in Fig.2.4) are almost identical to the ones obtained without the clustering stage. This indicates that the data reduction performed by the clustering comes with a negligible loss of information.

2.4.3 Numerical results on 500 segments

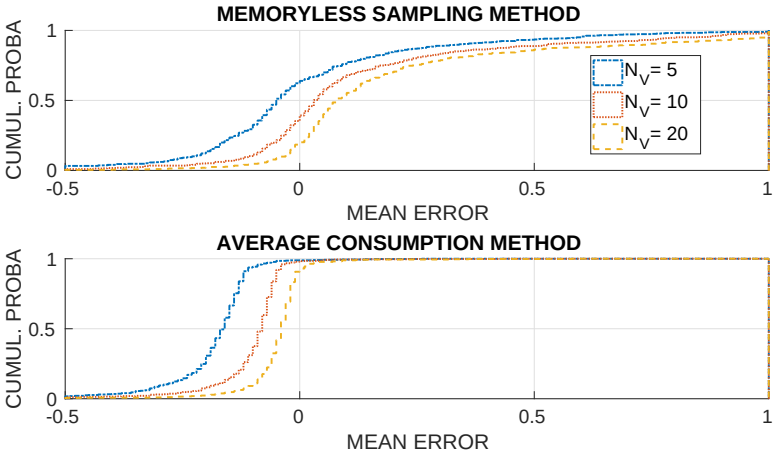
Now, we consider the entire road network of the city of Luxembourg, as implemented in LUST, with $\simeq 18000$ road segments. In order to study the influence of the different parameters while keeping reasonable computational times, we pick a test set of 500 random segments, for which we perform the clustering and compute the energy consumption with the Average and Memoryless methods. For each segment, we use the indicators defined in 2.3.2 to compare the computed energy consumptions to the Reference consumption. We discuss the relevance of the models based on the distributions of these indicators on the set of segments. More precisely, we investigate the influence of the discretization of speed and acceleration, and the number of clusters. Unless specified otherwise, the simulations use a speed and acceleration discretizations of 20 steps, a time-frame of 10 minutes, with 2 clusters.

Influence of speed discretization

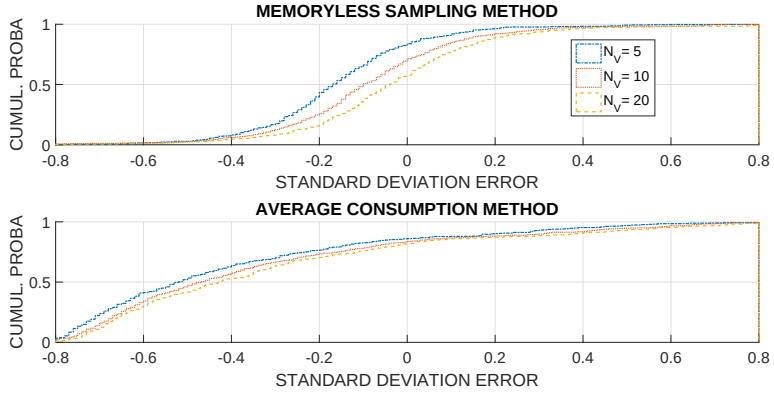
The speed discretization has a direct influence over the barycenters computed by the K-means, since it changes the support of the distribution obtained. We expect a finer discretization to give computed consumptions (Average and Memoryless Sampling methods) closer to the Reference ones. The obvious drawbacks are an increased cost of the barycenter computation and size of the distributions. We test $N_{\mathbb{V}} = 5, 10, 20$ steps for the discretization, the speed interval being $[0, 20]$ in m/s.

• Relative mean and standard deviation errors
We begin with the distribution over the 500 test segments of the mean and standard error (both relative). Fig. 2.6a and 2.6b (upper graphs) show the errors between the Reference energy consumption and the Memoryless Sampling method. The mean and standard error both appear to be reasonably well centered around 0. We also observe that finer discretizations of the speed clearly improve the standard error, possibly due to a better reconstruction of the travel times. On the other hand the mean error is shifted towards positive values, and is indeed smaller for 10 steps than 20.

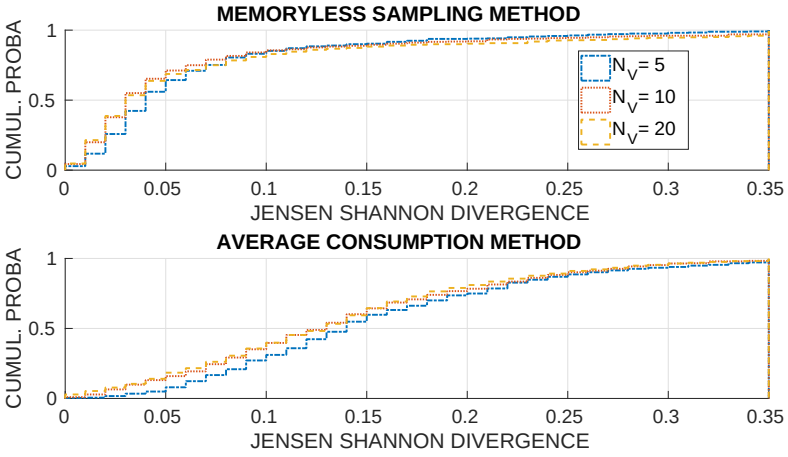
Fig.2.6a and Fig.2.6b (lower graphs) compare the Average method consumptions to the Reference. Here the mean error is almost always negative, and the standard error is also negative for 80% of the segments. This strong unbalance towards the negative indicates that the Average method tends to underestimate the consumption. Increasing the speed discretization reduces the mean error, but does not really improve the standard error. A possible explanation is the fact that the travel time for each vehicle is taken from the simulation, thus the speed discretization has no effect on it.



(a) Cumulated distribution of relative Mean error.



(b) Cumulated distribution of relative Standard deviation error.



(c) Cumulated distribution of Jensen-Shannon divergence.

Figure 2.6 – Speed discretization.

- Jensen-Shannon Divergence

Fig.2.6c shows the distribution over the 500 segments of the Jensen-Shannon divergence. Upper graph is for the Memoryless Sampling versus Reference, and lower graph is for Average method versus Reference. We observe that the JS divergence is much smaller

overall for the Memoryless Sampling than the Average method, with distributions more concentrated towards zero. For both methods, increasing the speed discretizations from 5 to 10 steps improves the JS divergence, while 20 steps yield very little additional gain.

Influence of acceleration discretization

Now we study the discretization of the acceleration. As for the speed, this parameter influences the support of the (v, a) distributions, and therefore the K-means clustering. We want to know if finer discretizations of a give more accurate energy consumptions for the Average and Memoryless Sampling methods. We test $N_{\mathbb{A}} = 5, 10, 20$ steps for the discretization, the acceleration interval being $[-4.5, 4.5]$ in ms^{-2}.

• Relative mean and standard deviation errors
Fig.2.7a shows the mean error distribution for the two methods, with different discretizations of a. We observe that for both the Memoryless and Average methods, going from 5 to 10 steps gives a significant improvement, while 20 steps is similar to 10. With a sufficient discretization, the mean error is extremely good for the Average method. The Memoryless method, on the other hand, tends to slightly overestimate the consumption.

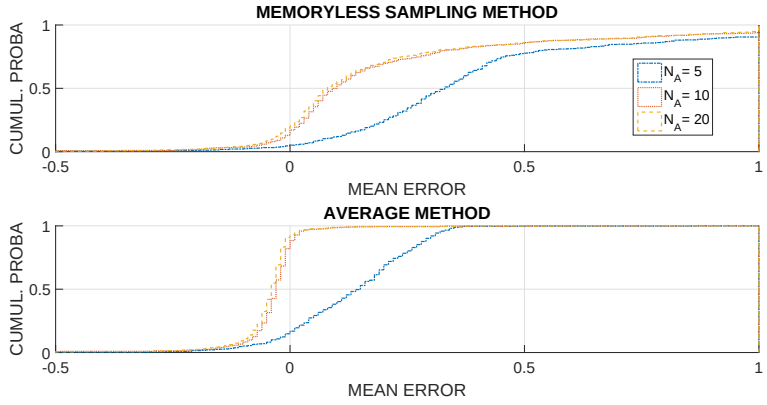
Fig.2.7b shows the standard error distributions. As observed for the speed discretization, error for the Memoryless is well balanced while Average method has mostly negative standard errors. Finer discretizations of a seem to give no improvement for the standard error. This may be due to the fact that acceleration has no influence on the travel time for either method, unlike speed which is used to reconstruct the travel times in the Memoryless method.

• Jensen-Shannon Divergence
Fig. 2.7c shows the distribution of the Jensen-Shannon divergence when varying the discretization of a. Like in the speed discretization study, we observe that the JS divergence is much smaller overall for the Memoryless Sampling method. Once again, for both methods increasing from 5 to 10 steps improves the indicator, while 20 steps give no additional benefit.

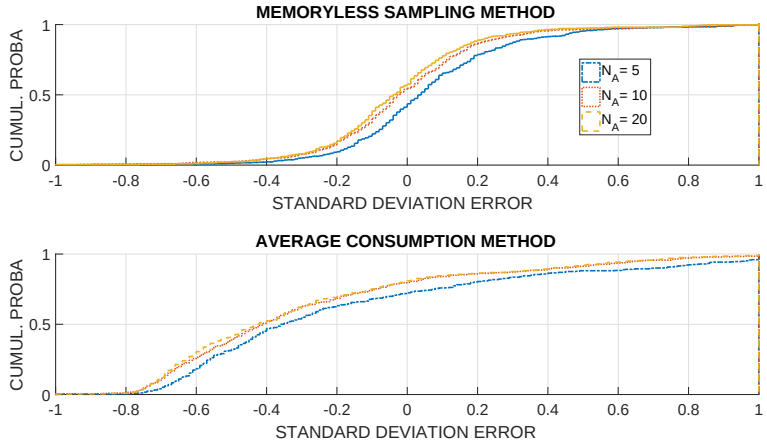
Influence of the number of clusters

Our traffic model uses clustering techniques to reduce the size of the traffic data, while retaining the useful information. One would expect some kind of trade-off, where using a larger number of clusters would keep more information at the expense of data size.

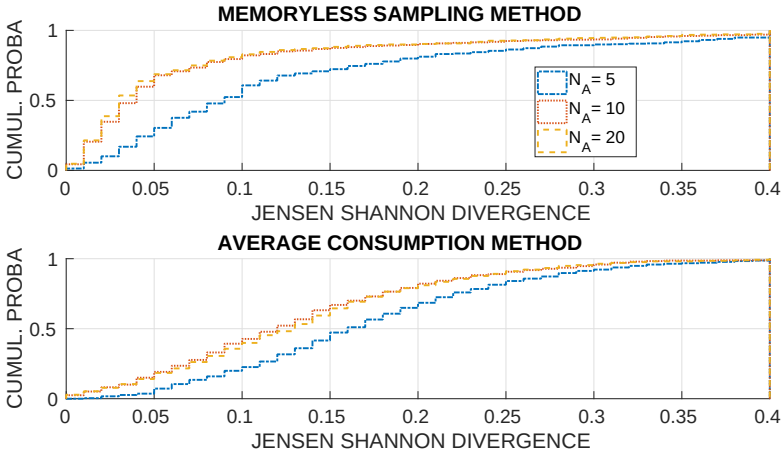
For this particular segment, simulations with $k = 2, 3, 4$ clusters give almost identical consumption distributions. Note that these simulations were run for a time-frame $\Delta t = 1h$, due to the increased computational cost for 3 and 4 clusters. It remains to be seen whether a smaller time-frame would benefit more from a higher number of clusters. Also, the ideal number of clusters is most likely segment-dependent. Experiments with real data in Sec. 2.5 give more insight into the impact of the number of clusters.



(a) Cumulated distribution of relative Mean error.



(b) Cumulated distribution of relative Standard deviation error.



(c) Cumulated distribution of Jensen-Shannon divergence.

Figure 2.7 – Acceleration discretization.

Summary and performance analysis

Generally speaking, increasing the speed, acceleration, time discretization and the number of clusters will improve the accuracy of the whole approach. However, this comes

at a cost, and some parameters have more impact than others. The detailed influence of the method parameters is as follows:
- speed/acceleration discretization: greater N_V improves the Memoryless method indicators except mean error, and the Average method except std error. Greater N_A improves all indicators except std error for Average method.
- time-frame Δt: seems to bring little improvement to both methods with this data set.
- number of clusters k: for this data set, seems to have little effect overall as well.

The drawbacks of each of these parameters are:
- finer speed/acceleration discretizations come at the cost of increased data occupancy (before and after clustering) and higher CPU time for the clustering.
- smaller time-frames increase data occupancy before clustering, clustering CPU time, and the risk of not having enough data on the segments for the distributions to be meaningful.
- a larger number of clusters leads to a higher data occupancy after clustering, and clustering CPU time.

Overall, the validation indicates that the Memoryless method has an acceptable mean error and a good std error, while the Average method has a very good mean error but a bad std error. The JS divergence is harder to interpret but is lower for the Memoryless method (with half of the distribution below 0.025) than the Average method (half of the distribution below 0.125).

At the core, we are interested in making use of the statistical information from the traffic distributions. In this respect, the Average method appears too limited, with a bad std error regardless of the parameters choice. Therefore, in the following sections we will focus on the Memoryless method.

For the 1-day simulation on the whole city of Luxembourg, the raw data corresponding to the 18322 segments take up to 5.7GB. Table 2.1 indicates the data size before ('HIST') and after ('BARY') clustering, as well as the CPU time for the clustering, for a 10% subset of the 18322 segments. For the whole dataset, with a discretization $N_V = N_A = 10$ and a time-frame $\Delta t = 10min$ the histograms for the (speed,acceleration) distributions are computed in less than 15min and amount to 1.2GB. Setting $k = 4$, the clustering step takes 19h and the distributions for the barycenters occupy 59MB. Taking the histograms already reduces data occupancy by 79%, and clustering pushes it to 99% in total. More generally, the data reduction done by the histograms is related to the speed/acceleration discretization, while the clustering further reduces data size according to the ratio $k/\Delta t$. Memory usage during the computations was about 100MB and posed no difficulties. It should be noted that the clustering is an offline step that only has to be done once. Also, it is independent for all segments and may benefit heavily from parallelization.

In the end, we observe that the range of parameters that give a reasonable trade-off between accuracy, data size and computation times is rather small, which limits the interest of trying sophisticated techniques to deduce some 'optimal' settings. The number of clusters k is discussed again in 2.5. As for the N_V, N_A discretization, possible improvements could be for instance the kind of adaptive exploration described in [93], provided

$N_{V\|A}$	Δt	k	HIST	BARY	CPU
5	10 min	4	49 MB	1.5 MB	25min
10	10 min	4	127 MB	5.6 MB	1.9h
15	10 min	4	256 MB	13 MB	7.9h
10	5 h	4	7 MB	5.6 MB	9min
10	1 h	4	30 MB	5.6 MB	30min
10	10 min	4	127 MB	5.6 MB	1.9h
10	1 min	4	661 MB	5.6 MB	7.5h
10	10 min	2	127 MB	2.8 MB	55min
10	10 min	3	127 MB	4.2 MB	1.4h
10	10 min	4	127 MB	5.6 MB	1.9h
10	10 min	10	127 MB	14 MB	4.8h

Table 2.1 – Influence of parameters on CPU time and data size (HISTograms before clustering and BARYcenters after). Dataset is 10% of the total 18322 segments of the Luxembourg scenario, with raw size about 480MB.

the overall cost of the exploration does not outweight the cost of simply taking a sufficient fixed discretization.

2.5 Results using real data

Thanks to floating car data collected by the smartphone application Geco air [40], we were able to test our method on a real-life scenario. We focused on a portion of the A7 highway near Lyon, France, which is known to be regularly used by commuters. For our analysis, the traffic data collected during the working days of the last two years were aggregated as they were recorded over one day, reasonably assuming that the data share similar traffic patterns. The speed measurements were then divided into 10-minutes timeframes. The Memoryless method is applied to the 1632 collected speed profiles. The discretization of the (v, a) space is limited at a 10×10 grid to reduce the computation time. One of the main goal of our method is to represent traffic with a small number of (v, a) distributions, thus we choose to explore the traffic representation with a number of clusters between 1 and 10. As we did with the data from SUMO, we will compare the consumption distribution of the Memoryless method C_{MSM} to the reference consumptions C_{Ref}.

2.5.1 Clustering impact

The choice of the number of clusters k is related to a trade-off between data size, computational time and accuracy of the consumption estimate. A small number of clusters will reduce the data occupancy and CPU time, with the risk of a coarser consumption estimate. A larger k will give consumptions closer to the unclustered raw data, at the cost of increased data size and CPU time. Fig. 2.8 shows the mean and standard deviation errors between the consumption distributions C_{Ref} and C_{MSM}, for a number of clusters $k \in 1, \cdots, 10$, as well as the unclustered case (dashed line). We see that the mean error is below 5%, and further decreases below 2% for $k = 2, 3, 4$ clusters, while larger k have

larger errors. Concerning the standard deviation, it is below 0.4 and appears globally decreasing with k. The few quirks in the curve may be due to some lingering sensitivity of the barycenter computation to the initialization. In the following, we set $k = 4$ clusters since this value gives a good trade-off. The discussion in 2.5.3 shows the link between the clusters and the traffic conditions, and sheds some light on why a small number of clusters seems optimal, since typical traffic conditions likely encompass only a limited number of situations (fluid, jammed, etc).

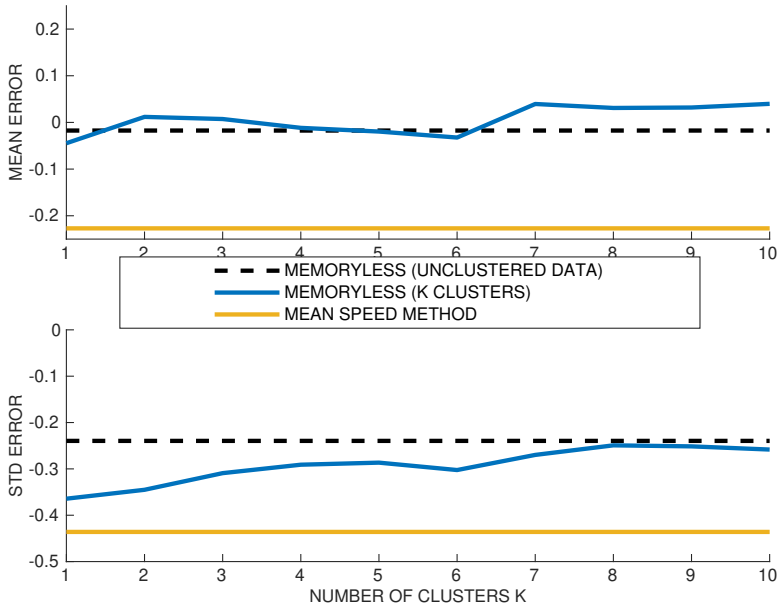


Figure 2.8 – Memoryless sampling method. Relative mean error and standard deviation for different numbers of clusters k. Errors for the unclustered data and the mean speed method are indicated as well. The value $k = 4$ seems to be a good choice in terms of errors and computational cost.

2.5.2 Mean speed method

Furthermore, we decide to compare our statistical model based on (v, a) distributions, to a simpler approach using only mean speeds, available from tools such as HERE or Google Maps. Noting $\bar{v}_{s,t}$ the mean speed for a given pair (segment,timeframe), we compute a basic estimate of the energy consumption C_{AVGS} with a constant speed and null acceleration

$$C_{AVGS} = \frac{P(\bar{v}_{s,t}, 0)L}{\bar{v}_{s,t}}. \tag{2.5.2.1}$$

Taking our set of (v, a) profiles from section 2.5, we recompute the mean speed information. We show on Fig. 2.9 the probability distributions for the reference consumption, memoryless method and mean speed method. We observe that the memoryless results are closer to the reference, with relative errors ($\varepsilon_{mean} = 4.6\%, \varepsilon_\sigma = 24.4\%$) better than ($\varepsilon_{mean} = 22.7\%, \varepsilon_\sigma = 43.6\%$) for the mean speed method.

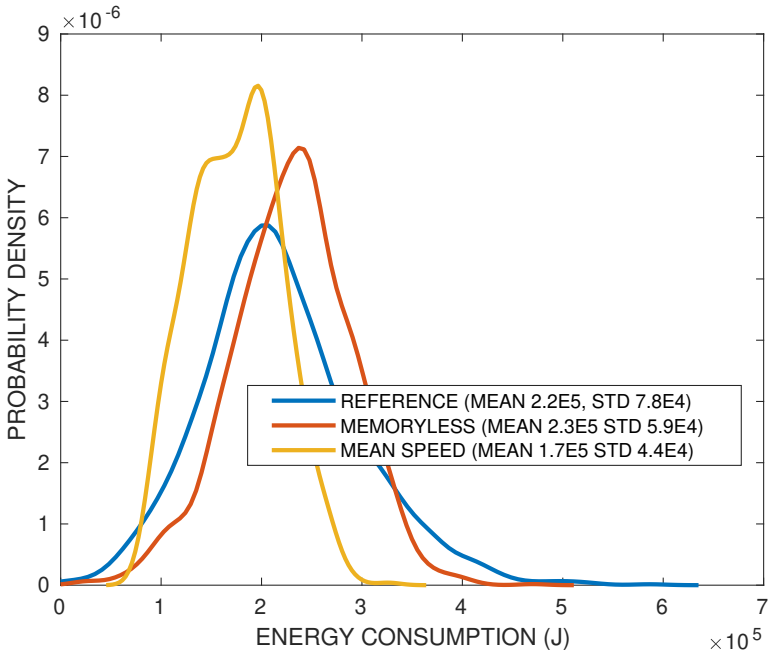


Figure 2.9 – Probability distributions of energy consumption. Reference (blue) vs Memoryless sampling method (red) and mean speed method (yellow). The mean and std are indicated for the 3 discrete distributions.

2.5.3 Analysis of the 4-cluster case

In the following we focus on the 4-cluster case that seems to offer a good trade-off between model size and accuracy. On Fig. 2.11 we plot the subset of speed profiles belonging to each of the 4 clusters. We observe that these speed profiles do appear rather similar in each cluster. Clusters 1 to 3 correspond to relatively smooth traffic conditions, with little speed variations, 2 being the fastest, followed by 1 then 3. Cluster 4, on the other hand, obviously corresponds to a traffic jam situation, with large variations in speed and frequent drops to null speeds.

Fig. 2.10 represents the same 4 subsets of profiles for each cluster in the (v, a) space. This representation confirms that the (v, a) distributions are quite distinct for each cluster. The level sets on each graph correspond to the distribution of the barycenter of the cluster, and we observe that the barycenters coincide rather well with the profile subsets. Furthermore, we can interpret the clusters in terms of traffic conditions, as summarized on Tab. 2.2.

Cluster	Mean Speed	Speed Spread	Acceleration	Interpretation
1	Medium	Important	Small	Normal
2	High	Low	Small	Fluid
3	Low	Low	Small	Dense
4	Very low	High	Well spread	Traffic jam

Table 2.2 – Traffic Interpretation

In order to see whether the traffic interpretation is coherent with reality, the clusters

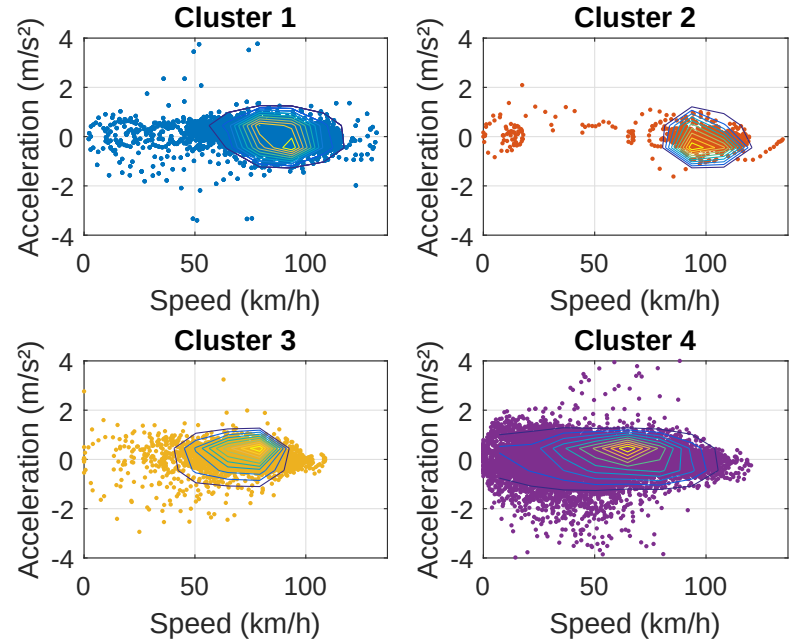


Figure 2.10 – Real distributions compared to their cluster.

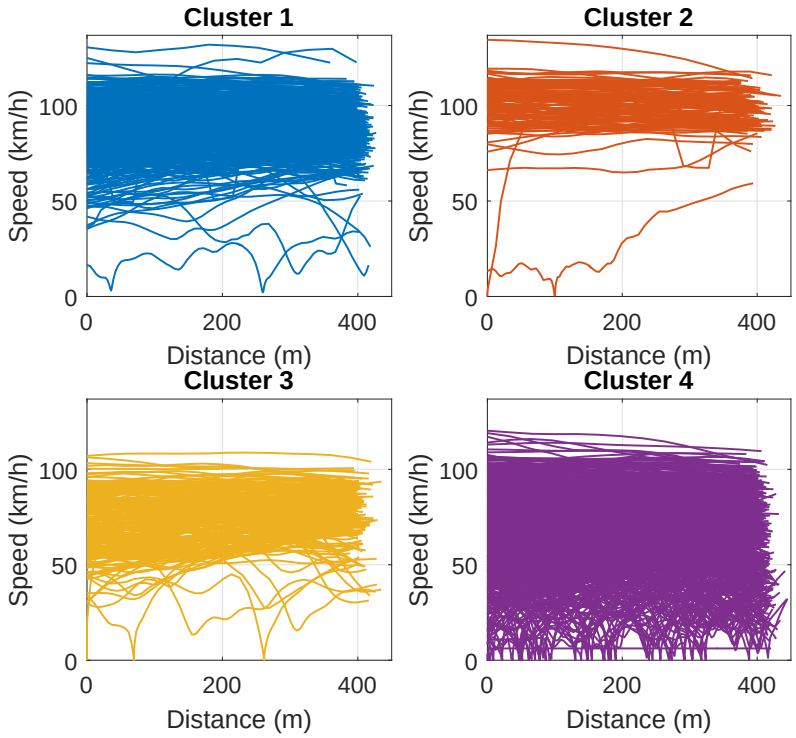


Figure 2.11 – Speed profiles associated with the clusters.

associated with each time-frame were plotted in Fig. 2.12, in which the grey portions correspond to the time-frames with not enough data for the analysis. We can observe

that the cluster 3, corresponding to a dense traffic, appears in the morning between 7am and 9:30am. The cluster 4, traffic jam, is present essentially between 4:30pm and 8pm. Cluster 1, normal traffic conditions, during the day and cluster 2, fluid conditions, during the night. Overall, this traffic pattern seems very consistent with reality, exhibiting two peak hours in the morning and evening, typical of commuting behavior.

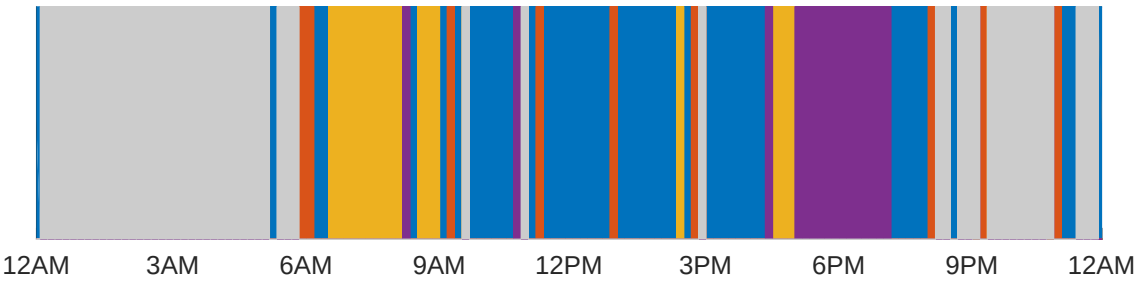


Figure 2.12 – Traffic clusters according to the time of day.

2.5.4 Summary

The analysis conducted on real data confirms that representing traffic by means of (v, a) distributions is effective not only to estimate the energy consumption distributions in different portions of the road network but also to identify different traffic conditions. Also, clustering proves to be an effective method to reduce data occupancy.

The cluster stage fits its role, allowing us to keep a reasonable data size while retaining most of the useful information from the original set of (v, a) profiles.

2.6 Conclusions

In this chapter, we have presented a new approach to use traffic data to predict the energy consumption of vehicles. The key point is to consider the (v, a) data in a statistical sense without the temporal aspect, coupled with a decomposition of the road network into a collection of small segments, based on topological aspects.

Numerical experiments carried out with traffic data generated by the traffic simulator SUMO indicate that our approach is able to reconstruct the distribution of the energy consumption over a set of vehicles. We introduce two methods to compute the energy consumption, called Average and Memoryless Sampling methods. The Memoryless Sampling method gives a more accurate estimate of the distribution of energy consumptions, according to indicators such as std error and Jensen-Shannon divergence.

We also investigate the influence of several parameters such as the (v, a) discretization, length of time-frame for data aggregation, and number of clusters for the data reduction.

The analysis on real data shows that the Memoryless Sampling method performs better than a more basic approach based on mean speed only. Another interesting point is that the clusters are consistent with the traffic conditions.

Possible future works include a second level of clustering, creating clusters of road segments with close traffic conditions, and variants of the Memoryless Sampling method

that would retain some temporality (e.g. Markov). Another direction could be to investigate the possible coupling of the presented model with approaches in flow management such as the SS-CTM in [48], since both use a spatial decomposition and account for stochastic aspects of the traffic.

CHAPTER 3

An Optimal local policy using traffic prediction

An article [55], based on this chapter and chapter 4, has been submitted to IEEE Transactions on Control Systems and Technology in 2019.

Contents

3.1 Introduction

Increased efficiency of HEVs derives from the on-board energy management system (EMS) which optimizes at each time instant the power split ratio between the two propulsion systems. A detailed review of the existing power management control algorithms for HEVs is offered by [63]. However, most of the current EMS strategies are somewhat conservative and sub-optimal due to their lack of prediction capabilities of the actual driving conditions of the vehicle. In fact, driving behavior and traffic conditions have a major impact on the traction power demand and consequently on the EMS. Several recent studies attempt to precisely establish such a relationship between driving conditions and energy consumption for different types of vehicle powertrains [24, 36].

The advent of connectivity and the availability of large amounts of driving data is favoring the transition towards predictive EMS strategies, which can further improve energy efficiency of HEVs by more effectively taking into account road traffic externalities [65]. Such predictive strategies need to have an estimate of the required power for traction along the vehicle trajectory, based on information about traffic conditions, road signalization and road grade. Data-based driving behavior and traffic models are typically based on historical information about traffic conditions on the different portions of a road network. Speed and acceleration probability distributions and their statistical properties are generally used to represent driving behavior [12, 61] and to establish speed predictors. Those predictors either combine deterministic and stochastic approaches [10, 92], or are fully based on stochastic processes such as Markov chains [85, 49, 67], or are determined through independent and identically distributed (i.i.d.) sequences [56]. These probability distributions are often obtained from standard driving cycles [68, 92] or real driving data [47, 49, 39, 56]. Their ability in reproducing real driving conditions strongly affects the performance of predictive EMS.

Since stochastic processes are considered as an effective way of predicting driving behavior, the research on EMS for HEVs has put much effort in designing predictive and stochastic optimization strategies. Such strategies can be essentially grouped into offline and online optimization methods. The offline methods such as the Stochastic Dynamic Programming (SDP) are mostly used as a benchmark for the online methods, but seldom used in practical implementations due to the high computational cost [14]. The online methods, such as the stochastic Model Predictive Control (MPC), in turn, offer more practical computation times without significantly compromising accuracy and performance. Stochastic MPC methods offer a favorable framework for the online prediction of the driving behavior on a future receding horizon and are employed for the design of EMS for HEVs [45]. However, accuracy and optimality of the MPC methods strongly depend on the accuracy of the driving behavior model, namely the probability distribution generating the stochastic process, as well as the size of the prediction horizon, typically chosen in a trade-off between performance and computational burden.

In an effort to reduce the impact of the prediction and optimization horizon on the EMS performance, bi-level optimization strategies seem promising in improving performance from a global perspective thanks to their hierarchical structure [81]. The intuition behind this type of optimization strategy for the EMS of HEVs is that in the system there are slowly changing variables, such as the battery state of charge (SoC), and rapidly changing variables, such as torque and regime. Hierarchical MPC has been applied to the EMS problem for HEVs [84, 38] and proved compatible with real-time implementation.

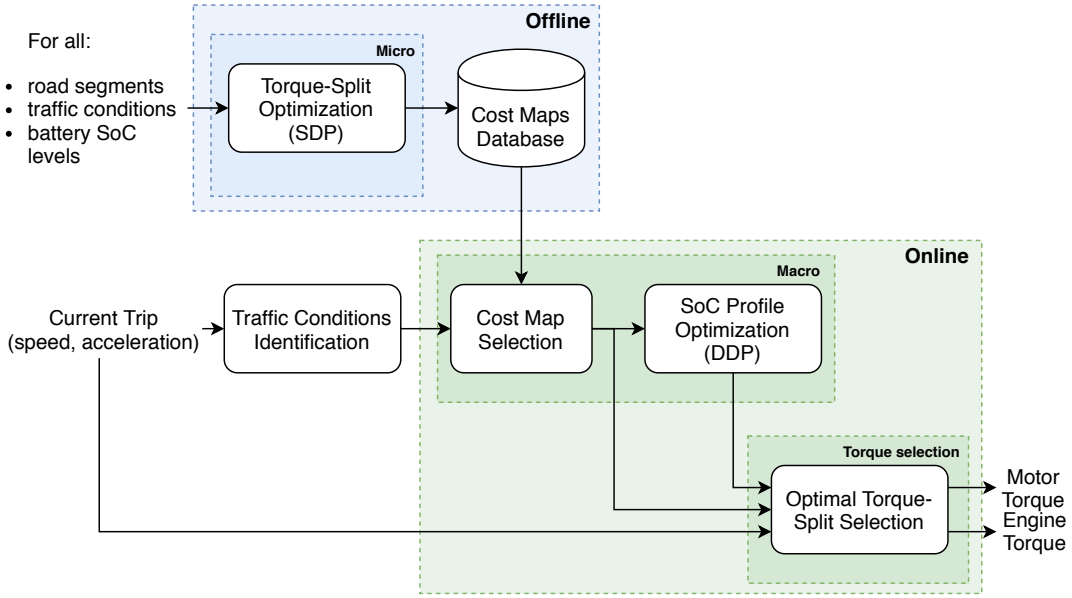


Figure 3.1 – Bi-level EMS proposed

However, hierarchical MPC still solves an online optimization problem thus requiring modeling simplifications and limited prediction horizon.

In this thesis, the proposed EMS strategy aims to combine the advantages of the infinite horizon optimization of the SDP and the real-time capability of the MPC. An illustrative scheme of the proposed approach is given in Fig.3.1. This chapter focus on the Torque-Split Optimization. The second part of the proposed EMS, the SoC Profile Optimization, is treated in chapter 4.

3.2 Modelling vehicles and traffic conditions

3.2.1 Hybrid vehicle model

Automobile companies have presented different architectures for hybrid electric vehicles. HEVs can be classified in four main types (Serial, Parallel, Combined, Complex) [15]. In the following we focus on HEVs with parallel design, where both the thermal engine and the electric motor can power the vehicle[1], see Fig.3.2. This type of design can use the engine in order to recharge the battery, at the cost of an increased fuel consumption. This option allows for optimization of the global consumption if some information is available on traffic conditions along the travel.

Neglecting the slope effect and knowing the speed v and the acceleration a, the torque T_w and the rotation speed ω_w of the wheel at each time t can be computed using the following formula (3.2.1.1):

$$T_w(v(t), a(t)) = (ma(t) + \alpha_2 v^2(t) + \alpha_1 v(t) + \alpha_0) r_w \tag{3.2.1.1a}$$

$$\omega_w(v(t), a(t)) = \frac{60 v(t)}{2\pi r_w} \tag{3.2.1.1b}$$

[1]Notice that the proposed strategy could be applied to every HEV architecture.

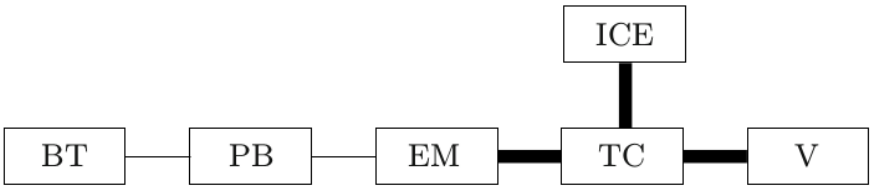


Figure 3.2 – Parallel hybrid configuration where ICE = internal combustion engine, BT = battery, PB = Power link, EM = electric motor, TC = torque coupler, V = vehicle. Bold lines indicate mechanical links, solid lines indicate electrical links.

with m the vehicle mass, r_w the wheel radius, and $\alpha_0, \alpha_1, \alpha_2$ coefficients of a quadratic approximation of the road-load force.

Then the torque required at the primary shaft T_{prim} and the rotation speeds of the engine ω_e and motor ω_m follow (3.2.1.2):

$$T_{prim}(v(t), a(t)) = \max\left(\frac{T_w(v(t), a(t))}{Pa_{ratio} Pa_{eff} G_R^i G_{Eff}^i}, T_{min}\right) \tag{3.2.1.2a}$$

$$\omega_e(v(t), a(t)) = \omega_w(v(t), a(t)) Pa_{ratio} G_R^i G_{Eff}^i \tag{3.2.1.2b}$$

$$\omega_m(v(t), a(t)) = \omega_w(v(t), a(t)) Pa_{ratio} G_R^i G_{Eff}^i R \tag{3.2.1.2c}$$

with G_R^i, G_{Eff}^i the gear ratio and efficiency, and Pa_{ratio}, Pa_{eff} the characteristics of the powertrain. Finally, R is the reduction ratio between the electric motor and the engine.

Neglecting losses due to the mechanical links, the torque of the engine and motor are linked through equation (3.2.1.3):

$$T_{prim}(v(t), a(t)) = T_e(t) + T_m(t) R \tag{3.2.1.3}$$

The consumption of the engine is modeled by a map $\hat{C}(\omega_e, T_e)$ obtained through experimental characterization, see Fig. 3.3. This map uses as inputs the torque request T_e and the rotation speed ω_e for the engine. Thanks to (3.2.1.1,3.2.1.2,3.2.1.3), we can express the consumption as a function of the electric motor torque T_m rather than T_e:

$$\begin{aligned} C(v(t), a(t), T_m(t)) = \hat{C}(&\omega_e(v(t), a(t)), \\ &T_{prim}(v(t), a(t)) - T_m(t) R). \end{aligned} \tag{3.2.1.4}$$

This reformulation allows us to take the motor torque T_m as the control of our system. Similarly to the engine map, we also have a motor map that gives the electrical power $\hat{P}_m$ required by the motor:

$$P_m(v(t), a(t), T_m(t)) := \hat{P}_m(\omega_m(v(t), a(t)), T_m(t)) \tag{3.2.1.5}$$

This power can be positive or negative, corresponding to a discharge (resp. charge) of the battery. We note C_{max} the maximum capacity of the battery and $SoC(t) \in [0, 1]$ its state of charge at time t. The dynamics of the state of charge can be written as follows:

$$\dot{SoC}(t) = \frac{1}{C_{max}} P_m(v(t), a(t), T_m(t)) \tag{3.2.1.6}$$

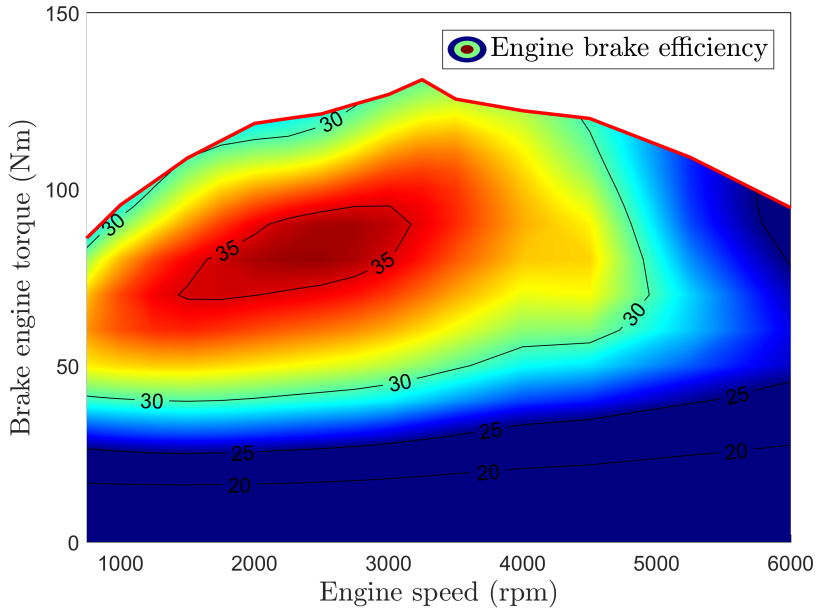


Figure 3.3 – Efficiency map of the engine

3.2.2 Probabilistic structure for traffic conditions

This section will introduce a formal framework for the representation of the traffic conditions using a stochastic framework. The results of the previous chapter indicate that taking the speed and the accelerations as drawn i.i.d. according to a probability distribution μ inherited from the past speeds and accelerations gives good approximation of the energy spent by the vehicle on a road segment.

The scenario space

Let $h > 0$ be the time step, $T > 0$ a horizon greater than the supremum of the exit time with probability 1. $N := T/h$ will be the number of time steps, and $\{t^k := kh, k = 0 \dots N\}$ the grid of time. A scenario ω is a function of time $k \to (v^k, a^k)$ corresponding to the speed and acceleration of our vehicle, assumed constant over the time interval and with (v^k, a^k) having values in a finite subset $G := \mathcal{V} \times \mathcal{A} \to \mathbb{R}^2$. Therefore a scenario ω represents a possible speed profile where $\omega^k = (v^k, a^k)$ represents respectively the mean speed and the mean acceleration of the vehicle over time interval (t^k, t^{k+1}).

We define the set of random events Ω as the set of scenarios. We assume the existence of a probability law $\mathbb{P}$ over Ω. As explained in e.g. [1, Ch. 7], this endows the event set with a Markov chain structure, whose transition probabilities, for (V, A) and (V', A') in G, are defined as

$$p^k(V, A, V', A') := \mathbb{P}\left[(v^{k+1}, a^{k+1}) = (V', A') | (v^k, a^k) = (V, A)\right] \tag{3.2.2.1}$$

and can be computed as follow :

$$p^k(V, A, V', A') := \frac{\mathbb{P}\left[(v^{k+1}, a^{k+1}) = (V', A') \text{ and } (v^k, a^k) = (V, A)\right]}{\mathbb{P}\left[(v^k, a^k) = (V, A)\right]} \tag{3.2.2.2}$$

Conversely, given probability transitions (such that the sum of probability transitions starting from a given node is equal to 1), the probability of a scenario is just the product of the corresponding sequence of probability transitions. In the memoryless probability transition model, the events $(v^{k+1}, a^{k+1}) = (V', A')$ and $(v^k, a^k) = (V, A)$ are independent, therefore

$$p^k(V, A, V', A') := \mathbb{P}\left[(v^{k+1}, a^{k+1}) = (V', A')\right] \tag{3.2.2.3}$$

We denote by $\mathcal{F}_k$ the associated filtration, which is the algebra generated by functions of $(\omega^0, ..., \omega^k)$.

Distance Process

A scenario defines a possible speed profile, as said in the previous section. Furthermore this speed profile describes the vehicle trajectory over the given road segment. Here $t \in \mathbb{R}^+$ denotes the time. The trajectories always start at time $t^0 = 0$, the time index k_t is defined such that $t \in [t^{k_t}, t^{k_t+1}]]$, i.e. $k_t \leq t/h \leq k_{t+1}$. Therefore the discrete trajectories of the speed and the acceleration are :

$$V(t) := v^{k_t}; A(t) := a^{k_t} \tag{3.2.2.4}$$

The distance covered by the vehicle at time t, denoted $D(t)$, is the solution of

$$\dot{D}(t) = V(t); D(0) = 0 \tag{3.2.2.5}$$

The exit time $t_f(\omega)$ corresponds to the earliest time for which the vehicle reaches the end of the road segment, namely

$$t_f(\omega) = \text{Min}\{t | D(t) \geq L\} \tag{3.2.2.6}$$

where L is the length of the segment. We recall the definition of a *stopping time* : a mapping $\tau : \Omega \to \{0, ..., N\}$ is a *stopping time*, if the event $\{\tau = k\}$ belongs to $\mathcal{F}_k$, for all k. The exit time t_f is a stopping time. Since the speed is random, the time spent on the road segment during the last time step is not equal to h but is proportional to the distance remaining to cover in order to complete the road segment.

Therefore the time spent by the vehicle on the segment, denoted $h^k(\omega)$ has value h if $k < k_{t_f}$ and $t_f(\omega) - k_{t_f}(\omega)h = \frac{L-D^k}{v^k}$ for $k = k_{t_f}(\omega)$. Thus $h^k(\omega) = \min\left(h, \frac{L-D^k}{v^k}\right)$.

Expected Consumption

As said in the previous section 3.2.1, the consumption is a function of the speed V, the acceleration A and the motor torque T_m. The decisions $T_m(\omega)$ are adapted to $\mathcal{F}_{k_t}$. We recall that a random variable depending on time (i.e., a collection of functions $y(t, \omega)$ for $\omega \in \Omega$) is adapted if, for all t, $y(t, .)$ depends only on realizations of ω up to time index k_t. In this context, the speed and acceleration at each time are given by ω. Therefore the decision T_m is chosen knowing the realisations of $(\mathbf{V}, \mathbf{A})$ until time k_t included. This classical framework, when the decision is chosen after the realization of the random variable is called Hazard Decision framework.

For a given scenario ω and decision $T_m(\omega)$, we define the associated cost

$$c(\omega, T_m(\omega)) := \sum_{k=0}^{k_f(\omega)} h_k(\omega) C(v^k, a^k, T_m^k(\omega^k),) \tag{3.2.2.7}$$

The corresponding expectation over the whole set of random events Ω can be written as

$$\mathcal{C}(T_m) = \mathbb{E}_{\mathbb{P}}\left[c(\omega, T_m(\omega)\right] = \sum_{\omega \in \Omega} \mathbb{P}(\omega) c(\omega, T_m(\omega)) \tag{3.2.2.8}$$

Finally, to a given decision map T_m can be associated a SoC process governed by

$$\dot{SoC}(t) = \frac{1}{C_{max}} P_m(v^{k_t}, a^{k_t}, T_m(v^{k_t}, a^{k_t})); SoC(0) = SoC_{init} \tag{3.2.2.9}$$

3.3 Optimal Control Problem under traffic conditions

Using the model presented in the previous sections, we are able to formulate an optimal control problem that will minimize the expected consumption under predicted traffic conditions. An analysis in term of error is also proposed.

3.3.1 State of Charge specifications

State of charge constraint

In order to avoid the trivial solution of a systematic maximal discharge, a constraint on the final state of charge SoC_f is added. This constraint can be an equality or an inequality. Equality constraints tend to be computationally tricky to handle in a stochastic framework, thus the inequality constraint is chosen in order to facilitate the numerical solving. It will ensure that the vehicle does not expend more than some allocated ΔSoC, namely

$$SoC(t_f(\omega)) \geq SoC_f := SoC(0) + \Delta SoC \tag{3.3.1.1}$$

Another constraint on the SoC needs to be considered, since the battery can never be below 0% or above 100%. This gives the running state constraint :

$$SoC(t) \in [0, 1] \tag{3.3.1.2}$$

Note that if the running state constraint is removed, the value for the problem depends only on ΔSoC, since the state of charge does not enter in the dynamics.

Feasibility of the final state of charge

Due to the physical limits of the motor and the engine, for each point of (V, A) in the grid G, there exists a possible maximal increase c_1 (resp. maximal decrease c_2) of the SoC. Then for the scenario w, a total variation $\Delta SoC(\omega)$ is not admissible if

$$\Delta SoC(\omega) \notin \sum_{k=0}^{k_{t_f}} [c_2(\omega^k), c_1(\omega^k)] \tag{3.3.1.3}$$

In practice, the maximal discharge c_2 is not likely to be a concern due to the final constraint on the SoC being an inequality. On the other hand, the maximal charge c_1 will make the problem infeasible as soon as there exists an infeasible scenario. In order to avoid this, we replace the final constraint on the SoC by a penalization term denoted $P(SoC_f, SoC_{t_f})$.

Monotonicity of the value functions

Intuitively, we expect that the value functions appearing in the problem satisfy the following 'monotonicity property': they are nonincreasing w.r.t. SoC_0 and nondecreasing w.r.t. SoC_f. We need the following hypotheses:

1. The consumption function C is a continuous and nonincreasing function of the torque T_m
2. The power function P_m is a continuous and nonincreasing function of the torque T_m
3. The final cost P is nonincreasing w.r.t. SoC_0 and nondecreasing w.r.t. SoC_f.

Theorem 1. *The value functions of the problem satisfy the above monotonicity property.*

Proof. The monotonicity w.r.t. SoC_f is a direct consequence of hypothesis 3). We next establish the monotonicity w.r.t. SoC_0. Consider a feasible policy T_m^k with associated state SoC^k (with k indicating the distance step). These are random variables, depending on the realization of speed and acceleration. Consider now the perturbed problem with perturbed initial state of charge $SoC_0' = SoC_0 + \varepsilon$, with $\varepsilon \geq 0$ and $SoC_0' \leq 1$. We consider a perturbed trajectory $(\hat{T}_m^k, \hat{SoC}^k)$ such that for all k, $\hat{SoC}^k$ is the smallest possible majorant of SoC^k. This obtained by forward induction: we choose the control T_m^{k-1} as the largest feasible one such that $\hat{SoC}^k \geq SoC^k$. By hypothesis 2) and Equation 5.2.2.9, the variations of state of charge of the perturbed trajectory are not greater than the original ones. Therefore thanks to hypothesis 1), we have a non greater consumption at each step k. By hypothesis 3), the final cost for the perturbed trajectory is not greater as the original one. The conclusion follows. □

In the following numerical section 3.4, we choose

$$P(SoC_f, SoC_{t_f}) = \lambda |SoC_f - SoC_{t_f}| \tag{3.3.1.4}$$

The choice of λ will be discussed in the following.

Heuristic cost

The value of the value function depends on λ, with the pitfall that setting a large value of λ will ensure the final SoC constraint but also strongly degrade the objective value. Therefore a heuristic to choose the suitable value of λ is proposed. The idea is to convert the missing final SoC into a corresponding amount of fuel. The mechanical energy that can be obtained for a litre of fuel is directly linked to the efficiency of the engine. As this efficiency dependent directly on the charging point, then the distribution μ can be used to compute an approximate marginal mean cost of the state of charge. Taking this mean cost as λ gives a meaningful order for the penalization, with respect of the consumption of the vehicle.

3.3.2 Problem statement

In order to model the uncertainty due to the traffic behaviour, the transition probability p^k is set as $\forall k, p^k(V, A, V', A') = \mu(V', A')$. We recall from chapter 2 that μ is a distribution representing traffic conditions.

So our problem $\mathcal{P}$ can be written as :

$$\min_{T_m} \quad \mathbb{E}_{\mathbb{P}}\left[c(\omega, T_m(t,\omega)) + P(SoC_f, SoC_{t_f})\right] \tag{3.3.2.1}$$

$$s.c\ \forall t, \quad \dot{SoC}(t) = \frac{1}{C_{max}} P_m(\omega^{k_t}, T_m(\omega^{k_t})) \tag{3.3.2.2}$$

$$\dot{D}(t) = V(t) \tag{3.3.2.3}$$

$$SoC(0) = SoC_{init},\ D(0) = 0 \tag{3.3.2.4}$$

$$s.c\ \forall t, \quad T_m(t) \in [T_{min}, T_{max}] \tag{3.3.2.5}$$

$$SoC(t) \in [0,1] \tag{3.3.2.6}$$

$$t_f = \min\{t | D(t) = L\} \tag{3.3.2.7}$$

Alternatively we can reformulate the above problem in a purely discrete time setting, dropping indexes s for the state variables:

$$\min_{T_m} \quad \mathbb{E}\left[\sum_{k=0}^{k_f} h^k C(\mathbf{V}^k, \mathbf{A}^k, T_m^k,) + P(SoC_f, SoC^{k_f})\right] \tag{3.3.2.8}$$

$$s.c\ \forall k, \quad SoC^{k+1} = SoC^k + \frac{h^k}{C_{max}} P_m(\mathbf{V}^k, \mathbf{A}^k, T_m^k) \tag{3.3.2.9}$$

$$D^{k+1} = D^k + h^k \mathbf{V}^k \tag{3.3.2.10}$$

$$T_m^k \in [T_{min}, T_{max}] \tag{3.3.2.11}$$

$$SoC^k \in [0,1] \tag{3.3.2.12}$$

$$h^k := \min(h_0, (L_s - D^k)/\mathbf{V}^k) \tag{3.3.2.13}$$

$$k_f := 1 + \max\{k, D^k + h_0 \mathbf{V}^k < L_s\} \tag{3.3.2.14}$$

$$SoC^0 = SoC_0;\ D^0 = 0. \tag{3.3.2.15}$$

Dynamic programming for the stochastic problem

A dynamic programming principle in discrete time is as follows. Clearly the value of the problem does not depend on time, so we look for $U(D, SoC) : [0, L_s] \times [0,1] \to \mathbb{R}$, value of the same problem one but starting with state (D, SoC). We have an exit time problem since the process stops when $D = L_s$.

For lighter notations, write $x := (D, SoC)$, and

$$x_+ := (D + h(D, \mathbf{V})\mathbf{V}, SoC + \frac{h(D, \mathbf{V})}{C_{max}} P_m(\mathbf{V}, \mathbf{A}, T_m)), \tag{3.3.2.16}$$

where $h(D, v) := \min(h_0, (L_s - D)/v)s$. Then

$$U(x) = \mathbb{E}\left(\min_{T_m \in \mathcal{T}(SoC, D, \mathbf{V}, \mathbf{A})} [h(D, \mathbf{V}) C(\mathbf{V}, \mathbf{A}, T_m) + U(x_+)]\right) \tag{3.3.2.17}$$

where the expectation is over $(\mathbf{V}, \mathbf{A})$ with law μ, with exit cost

$$U(L_s, SoC) = P_s(SoC, SoC_f). \tag{3.3.2.18}$$

Stochastic Dynamic programming

For the practical computations, the state space for (D, SoC) is discretize with steps $h_D := L/N_D$ and $h_{SoC} := 1/N_{SoC}$, where N_D and N_{SoC} are positive integers. The value function $U^h = U^h(x)$ is defined at gridpoints $x := (k_D h_D, k_{SoC} h_{SoC})$ with $0 \leq k_D \leq N_D$, $0 \leq k_{SoC} \leq N_{SoC}$ and extended to $[0, L] \times [0, 1]$ by the usual multidimensional linear interpolation formula, see e.g. [35, Ch.3], denoted by $\tilde{U}^h$. By $\mathcal{G}$ (resp. $\mathcal{G}_-$) we denote the set of gridpoints (resp. of gridpoints with $D < L_s$). We consider the 'approximate' dynamic programming principle

$$U^h(D, SoC) = \mathbb{E} \min_{T_m \in \mathcal{T}(SoC, D, \mathbf{V}, \mathbf{A})} \Big[h(D, \mathbf{V}) C(\mathbf{V}, \mathbf{A}, T_m) + \tilde{U}^h(x_+) \Big] \tag{3.3.2.19}$$

for all $x = (D, SoC) \in \mathcal{G}_-$, with exit cost as in (3.3.2.18). Then a classical Stochastic Dynamic Programming algorithm 2 can compute the value function over the whole state grid, taking T large enough.

Algorithm 2: SDP algorithm in Hazard Decision framework

for *t:T-1* $\rightarrow$ *0* **do**
  for $x \in \mathcal{G}$ **do**
    $U_t(x) = 0$;
    for $v, a) \in \mathrm{supp}(\mu)$ **do**
      $u = \infty$;
      for $T_m \in \mathcal{T}(SoC, D, \boldsymbol{V}, \boldsymbol{A})$ **do**
        $u = \min(u, h(D, v) C(v, a, T_m) + U_{t+1}(x_+))$;
      $U_t(x) = U_t(x) + u$;

Note that solving the problem 3.3.2.8 for a prescribed final condition SoC_f actually provides the cost map $\nu(SoC_0, SoC_f)$ for all SoC_0 points on the SoC grid.

Error due to state variables discretization

Since the velocities are non negative and have a positive expectation, the above formula implicitly expresses $U^h(D, SoC)$ as function of $U^h(D', \cdot)$ for $D' \in [D + h_D, L_s]$. So, (3.3.2.19) can be solved by backward induction over distances. For $D_k := k h_D$ and (D_k, SoC) in $\mathcal{G}_-$, set

$$U_k(SoC) := U(D_k, SoC); \;\; U_k^h(SoC) := U^h(D_k, SoC); \tag{3.3.2.20}$$

The corresponding value error is $W_k(SoC) := U_k^h(SoC) - U_k(SoC)$. Set

$$\hat{U}_k(SoC, \mathbf{V}, \mathbf{A}, T_m) := \tilde{U}(D_k + h(D_k, \mathbf{V})\mathbf{V}, \tag{3.3.2.21}$$

$$SoC + \frac{h(D_k, \mathbf{V})}{C_{max}} P_m(\mathbf{V}(t), \mathbf{A}(t), T_m(t)) \tag{3.3.2.22}$$

with a similar definition for $\hat{U}_k^h$, and set $\hat{W}_k := \hat{U}_k^h - \hat{U}_k$. From the above dynamic programming principles (3.3.2.17) and (3.3.2.19) we deduce that

$$\|\hat{W}_k\|_\infty \leq \mathbb{E} \sup_{SoC, T_m} \{|\hat{W}_k(SoC, \mathbf{V}, \mathbf{A}, T_m)|\}. \tag{3.3.2.23}$$

In the above supremum we take as always the expectation over $(\mathbf{V}, \mathbf{A})$ in the support of ρ_s, and the supremum over those $SoC \in [0,1]$ such that $(D_k, SoC) \in \mathcal{G}_-$, and over $T_m \in \mathcal{T}(SoC, D, \mathbf{V}, \mathbf{A})$.

Let the supremum be attained at $(\overline{SoC}, \bar{T}_m)$, which are functions of $(\mathbf{V}, \mathbf{A})$. Denote by $(\bar{\alpha}_i, \bar{x}_i)$ the coefficients and gridpoints of the corresponding linear interpolation (also function of $(\mathbf{V}, \mathbf{A})$ but we skip these arguments). Then

$$\|\hat{W}_k\|_\infty \leq \mathbb{E}\left(|\Delta_1(\mathbf{V}, \mathbf{A})| + |\Delta_2(\mathbf{V}, \mathbf{A})|\right), \tag{3.3.2.24}$$

with

$$\Delta_1(\mathbf{V}, \mathbf{A}) := \sum_i \bar{\alpha}_i (U^h(\bar{x}_i) - U(\bar{x}_i)); \tag{3.3.2.25}$$

$$\Delta_2(\mathbf{V}, \mathbf{A}) := \sum_i \bar{\alpha}_i (U(\bar{x}_i) - U(\bar{x}_+)). \tag{3.3.2.26}$$

We can interpret Δ_1 as a combination of previous errors at grid points, and Δ_2 as an interpolation error for U at x_+. Let $e_I(h_D, h_{SoC})$ denote a majorant of the interpolation error $|\Delta_2|$, so that we have

$$|\Delta_2| \leq e_I(h_D, h_{SoC}). \tag{3.3.2.27}$$

In the case when no interpolation in D is necessary, we denote the corresponding interpolation error by $e'_I(h_{SoC})$. We now estimate $|\Delta_1|$. Setting I (resp. J) for elements of gridpoints with distance index equal to (resp. greater than) k we get

$$\Delta_1 := \sum_{i \in I} \bar{\alpha}_i (U^h(x_i) - U(x_i)) + \sum_{i \in J} \bar{\alpha}_i (U^h(x_i) - U(x_i)) \tag{3.3.2.28}$$

so that setting $\beta := \mathbb{E} \sum_{i \in I} \bar{\alpha}_i$:

$$|\Delta_1| \leq \beta \|W_k\|_\infty + (1-\beta) \max_{k' > k} \|W_{k'}\|_\infty. \tag{3.3.2.29}$$

Observe that β represents the probability of having zero speed, and therefore is a given constant in $[0, 1)$.

Theorem 2. *The following error estimate holds: for all $(D, SoC) \in \mathcal{G}$, we have that*

$$\|U_k^h - U_k\|_\infty \leq \frac{L_s e_I(h_D, h_{SoC})}{(1-\beta) h_D}. \tag{3.3.2.30}$$

If in addition, whenever v belongs to the marginal (in speed) of ρ_s:

$$hv \text{ is a multiple of } h_D, \tag{3.3.2.31}$$

then

$$\|U_k^h - U_k\|_\infty \leq \frac{L_s e'_I(h_{SoC})}{(1-\beta) h_D}. \tag{3.3.2.32}$$

Proof. It follows from the previous discussion that

$$\|W_k\|_\infty \leq e_I(h_D, h_{SoC}) + \beta \|W_k\|_\infty + (1-\beta) \max_{k' > k} \|W_{k'}\|_\infty. \tag{3.3.2.33}$$

Equivalently

$$\|W_k\|_\infty \leq \max_{k' > k} \|W_{k'}\|_\infty + \frac{e_I(h_D, h_{SoC})}{1-\beta}. \tag{3.3.2.34}$$

Since there are L_s/h_D steps, (3.3.2.30) follows. Finally, if (3.3.2.31) holds, we deduce from (3.3.2.30) that (3.3.2.32) holds. □

Remark 3. If U is Lipschitz w.r.t. SoC with constant L_{SoC}, then $e'_I(h_{SoC}) \leq L_{SoC} h_{SoC}$. The resulting error estimate is then of order h_{SoC}/h_D, which is similar to the standard error estimates in the case of a fixed horizon (where the 'exit variable' is replaced by time), see e.g. the appendix by Falcone in [5].

Additional error due to computation

Instead of the approximate dynamic programming (3.3.2.19), at each step of the backward induction over distance, what we actually solve approximately is the problem

$$U_k^t(SoC) = \mathbb{E} \min_{T_m \in \mathcal{T}(SoC,D,\mathbf{V},\mathbf{A})} \left[h(D_k, \mathbf{V}) C(\mathbf{V}, \mathbf{A}, T_m) + \tilde{U}^c(x_+) \right] \tag{3.3.2.35}$$

for all $x = (D_k, SoC) \in \mathcal{G}_-$, with exit cost as in (3.3.2.18); here we have replaced the discrete value U^h in the l.h.s. with the 'target value' denoted by U^t, and the 'future values' $\tilde{U}^h(x_+)$ with the 'computed values' (at gridpoints) denoted by $\tilde{U}^c(x_+)$; but note that while the computed values are given (at grid points) for distances greater than $D_k = h_D k$, their values at distance D_k correspond to U_k^t. For $k \in \{0, \ldots, N_D\}$, denote the corresponding error term estimate by e_k, so that

$$e_k \geq \max_{k' \geq k} \|U_{k'}^c - U_{k'}\|_\infty. \tag{3.3.2.36}$$

We may assume e_k to be non-increasing, with zero value for $k = N_D$. By arguments similar to those of the previous section we obtain that for $(D_k, SoC) \in \mathcal{G}_-$:

$$\|U_k^t - U_k\|_\infty \leq e_I + e_{k+1}. \tag{3.3.2.37}$$

Therefore, by the triangle inequality

$$\|U_k^c - U_k\|_\infty \leq \|U_k^c - U_k^t\|_\infty + e_I + e_{k+1}. \tag{3.3.2.38}$$

Next, we choose to solve (3.3.2.35) by value iterations, i.e., as the limit of the sequence $U_k^i(SoC)$, for $i \in \mathbb{N}$, defined by

$$U_k^i(SoC) := \mathbb{E} \min_{T_m \in \mathcal{T}(SoC,D,\mathbf{V},\mathbf{A})} \left[h(D_k, \mathbf{V}) C(\mathbf{V}, \mathbf{A}, T_m) + \tilde{U_k^i}(x_+) \right]. \tag{3.3.2.39}$$

The infimum is of course for each grid value of SoC, and again the tilde corresponds to the interpolation operator. The contraction factor of the corresponding fixed-point operator is easily seen to be at most β. We initialize U_k^0 with U_{k+1}^c. It follows that

$$\|U_k^i - U_k^t\|_\infty \leq \beta^i \|U_{k+1}^c - U_k^t\|_\infty. \tag{3.3.2.40}$$

So, if we perform i_k iterations at step k we get with (3.3.2.38) that

$$\|U_k^c - U_k\|_\infty \leq e_I + e_{k+1} + \beta^{i_k} \|U_{k+1}^c - U_k^t\|_\infty. \tag{3.3.2.41}$$

While β may be computed, U_k^t is unknown so that explicit estimates can be derived only in specific examples. Nevertheless the above inequality suggests that, in the absence of additional information, it may be wise to take i_k independent on k.

3.4 Numerical simulations

In this section, we present the results obtained after solving the problem 3.3.2.8. Numerical simulation have been conducted on a passenger vehicle from [22], for which the value of the parameters are presented in Table. 3.1. The traffic model was derived from actual traffic data obtained thanks to GecoAir on a specific segment of the A7 near Lyon, see Fig 3.4. First, we discuss the influence of the traffic state on the Value Function, and therefore the expected consumption on the segment. Then we focus on specific speed trajectory in order to observe the optimal policy compute with our model. Finally, a comparison with a deterministic optimization is presented.

m	r_w	α_0	α_1	α_2
$1190kg$	0.31725m	113.5	0.774	0.4212

i	1	2	3	4	5
G_{R^i}	3.416	1.809	1.281	0.975	0.767
G^i_{Eff}	1	1	1	1	1

Pa_{ratio}	Pa_{eff}	R
59/13	0.95	3.3077

Table 3.1 – Parameters used in simulations

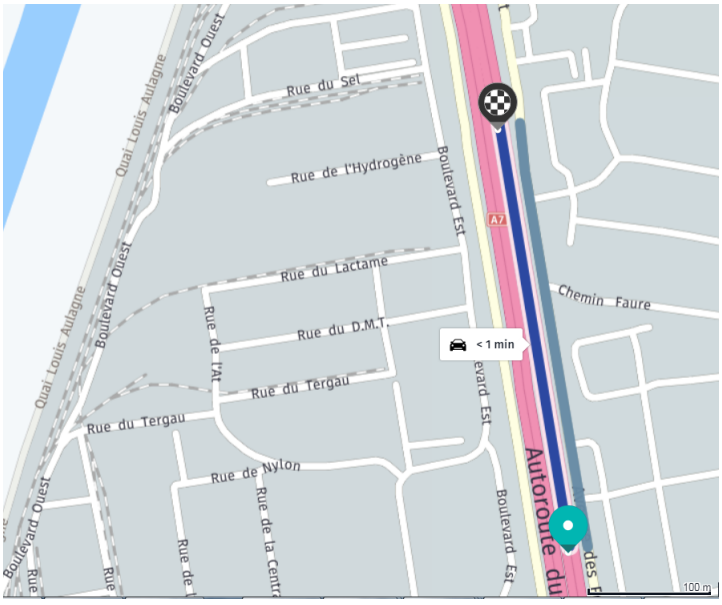


Figure 3.4 – Segment used for simulation

3.4.1 Value Function according to the traffic state μ

Thanks to the method proposed in [56], we have been able to determine a classification of traffic state.For k=4 clusters, the speed profiles contained in each are show on the

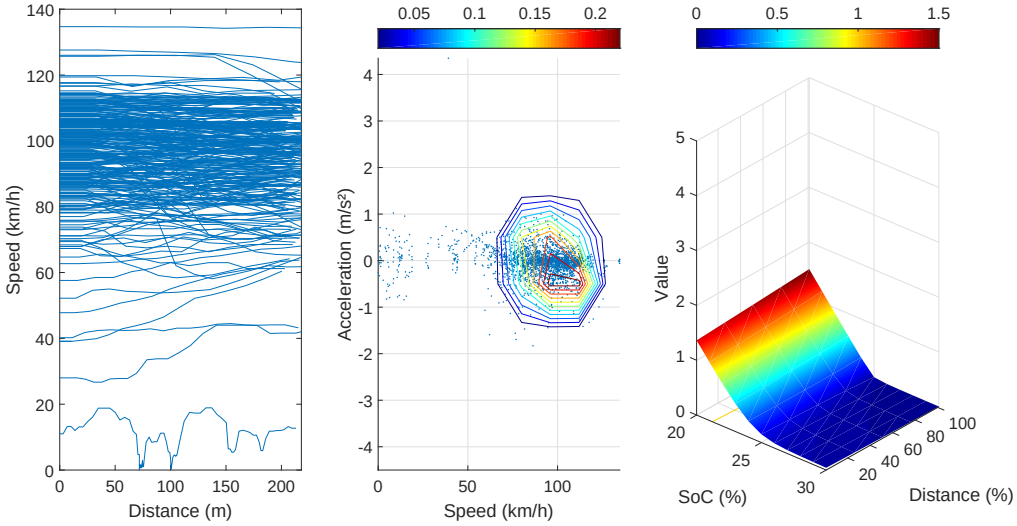


Figure 3.5 – Value Function obtained from cluster 1 on segment 9 which corresponds to a fluid traffic.

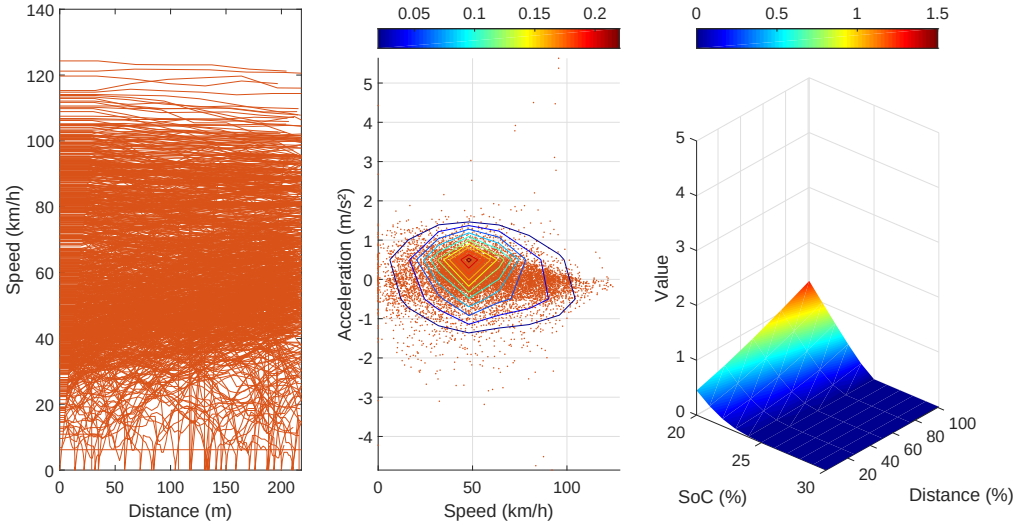


Figure 3.6 – Value Function obtained from cluster 2 on segment 9 which corresponds to a traffic jam.

left graph of Fig [3.5-3.8], while on the central graph the mean distribution of the corresponding cluster is shown. We can observe that cluster 1 contains essentially profiles with high speed thus the drivers are not constrained by each other, we deduce that cluster 1 represent a fluid traffic. On the opposite cluster 2 shows a large proportion of low speed profiles and even stops, indicating a traffic jam. Cluster 3 has a lower speed than cluster 1 but its mean speed is close to the speed limit (here 90km/h). Considering to the topological characteristics of the segment (2x2 ways), this could correspond to a large number of vehicle on the segment where drivers need to be careful about the others, and so slow down. The fourth cluster shows two peaks of speed around 50 km/h and 90 km/h, this can be explained by the fact that cluster 4 contains the time frame that are at junction between

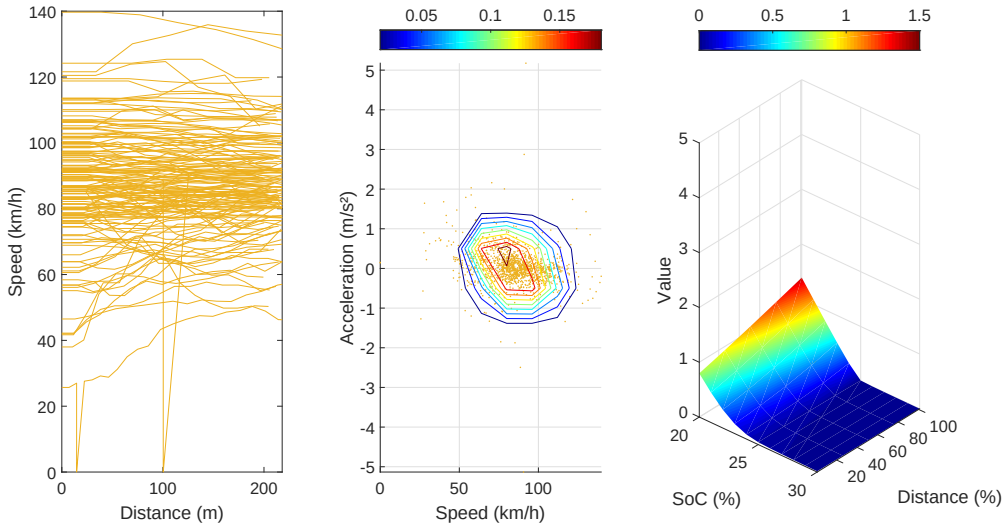


Figure 3.7 – Value Function obtained from cluster 3 on segment 9 which corresponds to a slightly slowed traffic.

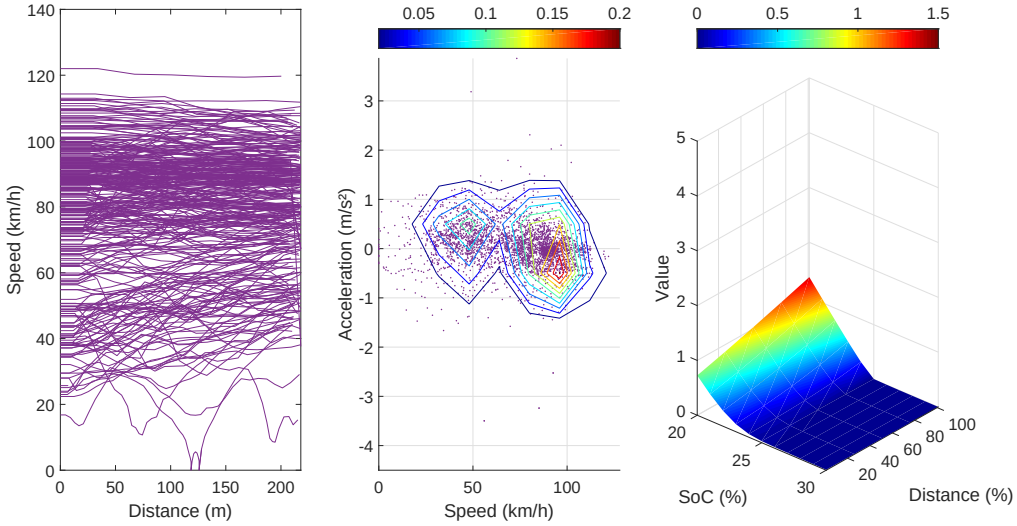


Figure 3.8 – Value Function obtained from cluster 4 on segment 9 which corresponds to an intermediate traffic state.

off peaks hours and peaks ones.

The Value Functions obtained for the different traffic state by solving the problem 3.3.2.8 with a final state of charge of 25% is shown on the right graph of Fig [3.5-3.8]. The shape of the value functions is quite similar, with some disctinctive features. First we observe a linear increase above a certain initial SoC. This behaviour is due to the final SoC penalization and this point is the minimal SoC which can satisfy the constraint. A first interesting point is the fact that this point depends on the traffic state. Indeed in high energy demanding traffic such as cluster 1, this point is very close or even above of our final SoC. On the opposite,traffic that allows regenerative braking leads to a point lower than the final SoC.

Fig [3.5-3.8] also shows that above a second value of the initial SoC, the expected consumption is stable. This point is the limit where it becomes possible to reach the final constraint while using only the electric motor.

To summarize, the value functions can be divided in three main regions according to the traffic conditions and initial SoC:

- The higher values using only the thermal engine
- The lower values using only the electric motor
- Between them, the hybrid driving

3.4.2 Optimal policy

In this section, the optimal policy obtained are shown for the same segment than the previous section, with a SoC_f equals to 25 %. We propose to compare the optimal control according to traffic with an initial SoC equal to 21 % , 23 % ,25 % and 27 %. This comparison is performed on cluster 1 and cluster 3, ans we interpret the controls in terms of operating mode the hybrid vehicle, i.e. the general functionnment of the power train.

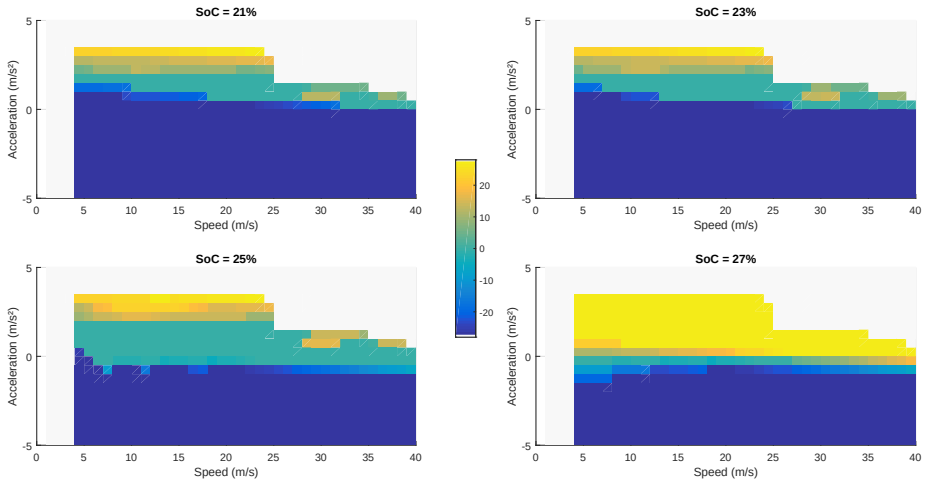


Figure 3.9 – Optimal Policy (electric torque T_m) obtained from cluster 1 on segment 9 which corresponds to a fluid traffic.

Cluster 1, shown on Fig. 3.9, corresponds to high speeds that require high energy. We observe that the electric motor is only used when it is mandatory in order to realize the demand of power when the state of charge is below 25%, and we use the ICE to recharge the battery. These modes are resumed on Fig. 3.10, the vehicle is in hybrid mode when the power asked is high (orange parts), or uses the ICE for medium power (red parts). When the power asked is low, the ICE is pushed in order to recharge the battery (black parts). The behaviour changes when the initial state of charge is greater than the final state of charge: for the 27% case the electric motor is always used and the engine is only used when it is mandatory. Fig. 3.10 shows that for low energy points the vehicle works only with the electric motor, and for greater energy the engine is used in order to complete the power asked.

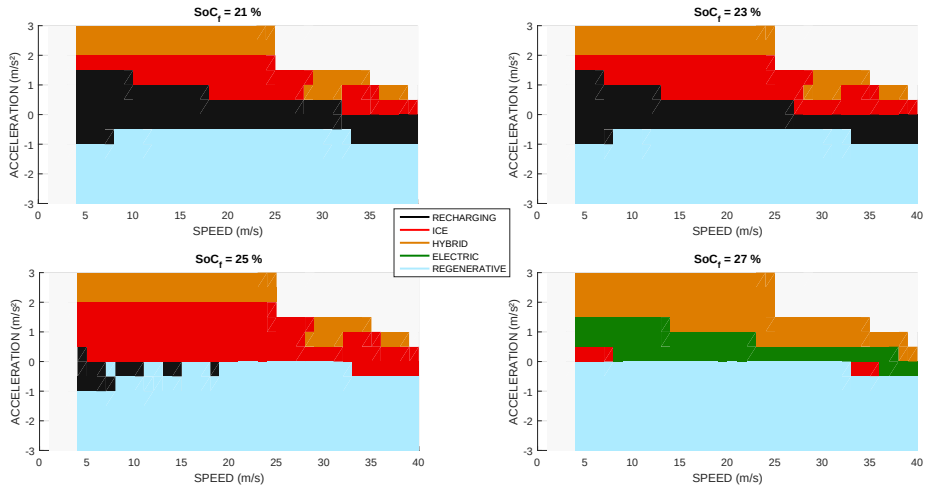


Figure 3.10 – Hybrid vehicle mode obtained from cluster 1 on segment 9 which corresponds to a fluid traffic.

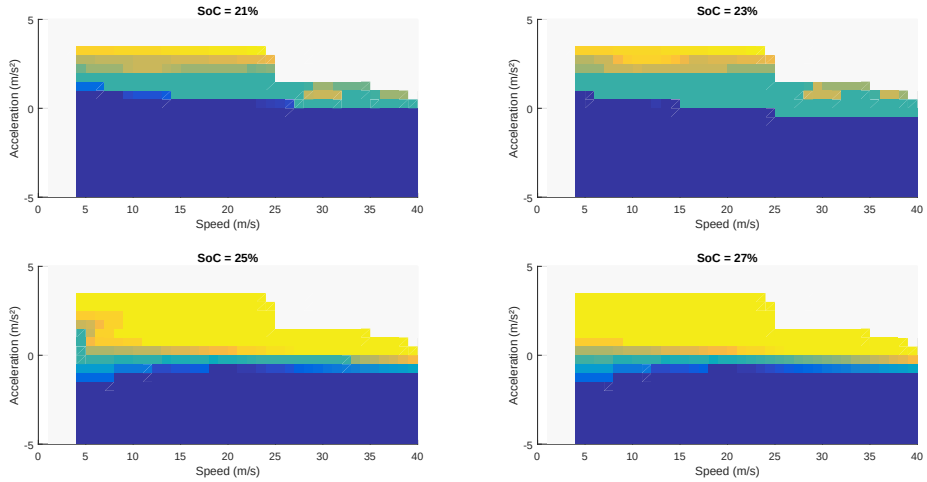


Figure 3.11 – Optimal Policy (electric torque T_m) obtained from cluster 3 on segment 9 which corresponds to a slightly slowed traffic.

Cluster 3 seems to show the same general behaviour with respect to the initial state of charge, except that since we are more likely to use regenerative braking, the electric mode appears for a lower SoC than cluster 1. This is shown on 3.12, where on the 25% case, the electric power part is significantly larger for cluster 3 than cluster 1. for a important part of the graph

On Fig. 3.13, the vehicle modes corresponding to the optimization with an isoSoc constraint are plotted for each cluster. We can observe that the electric motor is used when the cluster allows for regenerative braking, otherwise the engine is used in order to respect the constraint.

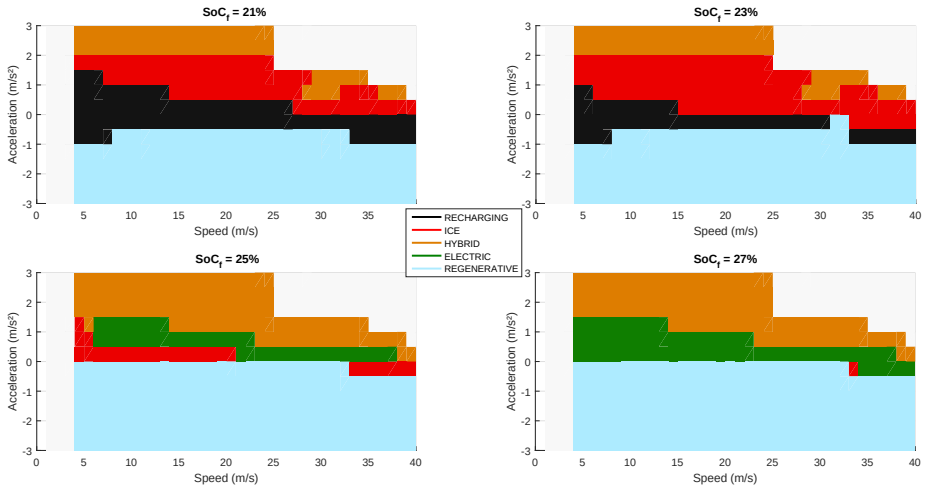


Figure 3.12 – Hybrid vehicle mode obtained from cluster 3 on segment 9 which corresponds to a slightly slowed traffic.

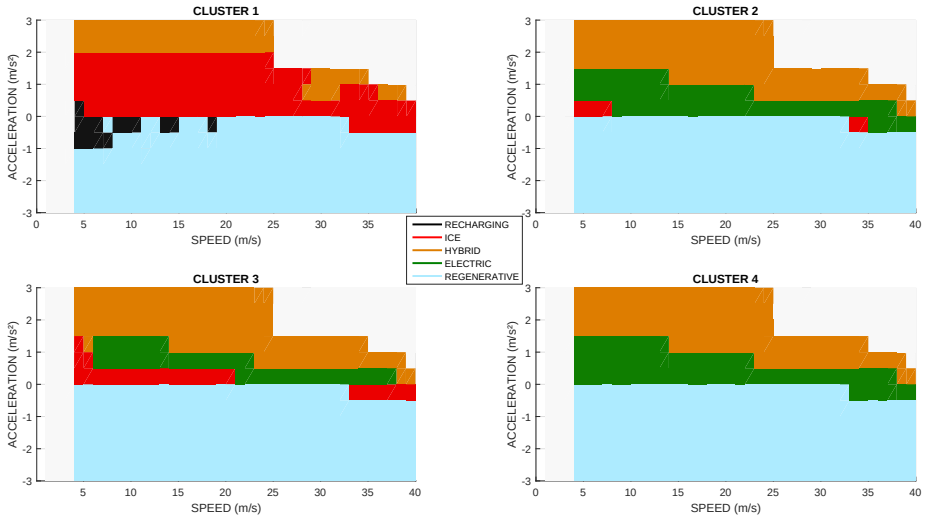


Figure 3.13 – Hybrid vehicle mode obtained from each cluster on segment 9 for a isoSoc constraint.

3.4.3 Speed trajectory

In this section, we observe the control obtained thanks to the predicted traffic and the control obtained if the exact speed was known. On the illustrative example shown on Fig 3.14, we observe that the control obtained thanks to the traffic prediction does not match the deterministic control. It seems that the control strategy from the stochastic optimization is more optimistic at the start and uses the battery more, hoping for future regenerative braking. This leads to a small overcomsuption at the end to meet the required final SoC. However, the consumptions of both methods are close, with a gap of less than 5%.

On Fig. 3.15, for a SoC constraint allowing q 4% discharge, the controls are quite close, with almost identical state of charge and consumption trajectory profile.

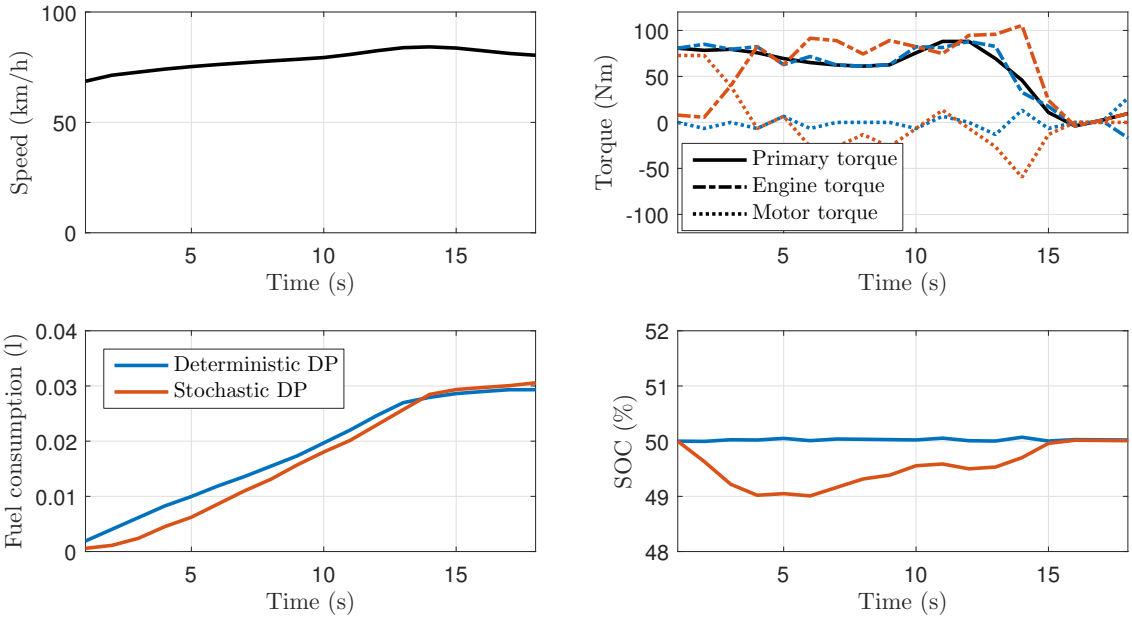


Figure 3.14 – A real speed profile from cluster 2 on segment 10 for a isosoc constraint

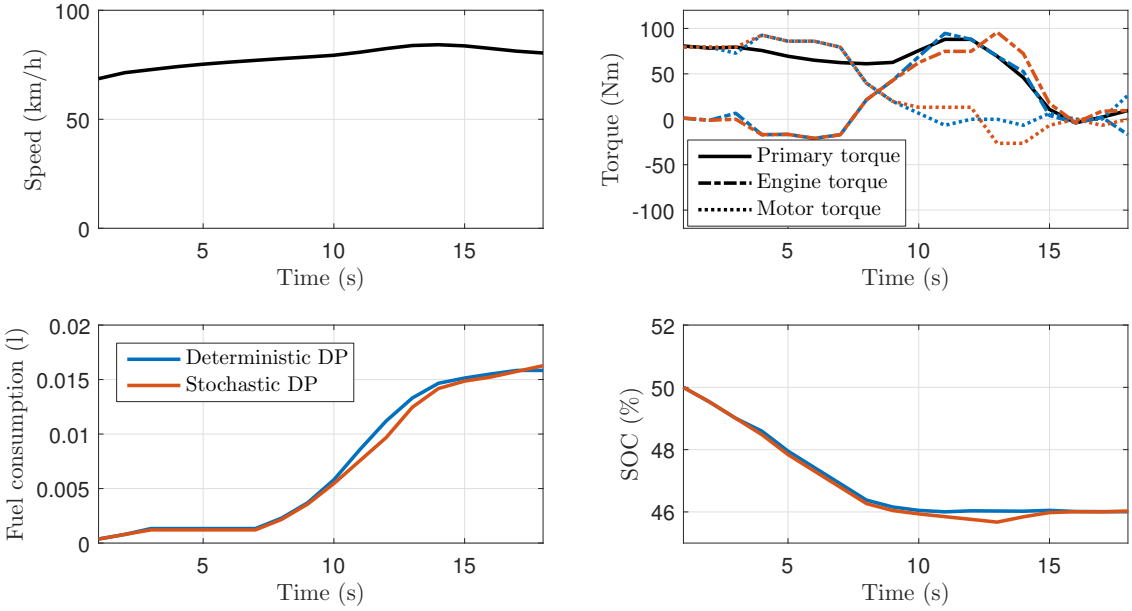


Figure 3.15 – A real speed profile from cluster 2 on segment 10 when depleting the battery of 4%

3.4.4 Estimation of the overconsumption

In this section, the consumption resulting from the stochastic optimal control is compared to the deterministic method. We recall that the deterministic problem gives the minimum consumption reachable with the vehicle knowing all the information.This comparison will therefore give an idea of the overconsumption introduced by the stochastic aspect of the traffic conditions.

On Fig.3.16, the consumption obtained thanks to the traffic distributions is relatively similar to the deterministic one, with a slight overconsumption. We can observe the slight

overconsumption of our methods. The mean of both methods have been plotted on the figure, and the relative mean error is close to 5%. This indicates that the stochastic method handles the traffic conditions with a reasonable overconsumption.

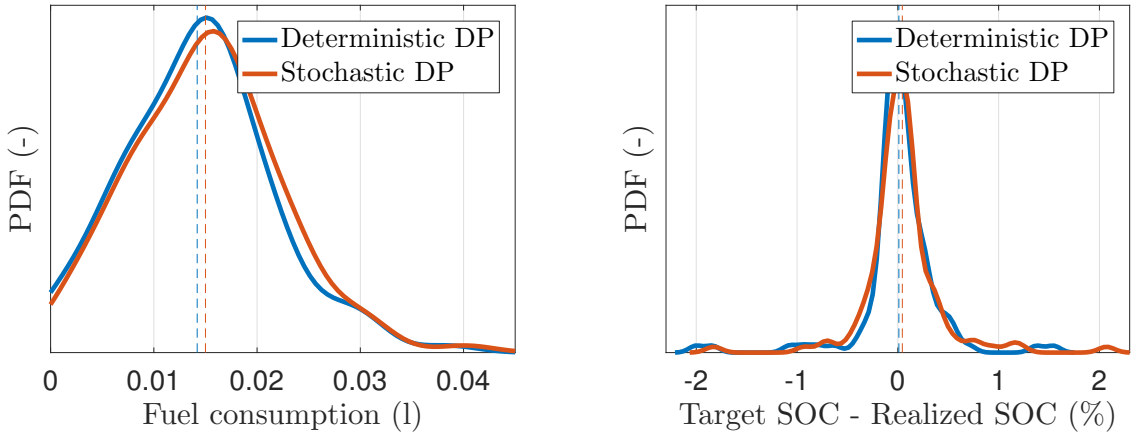


Figure 3.16 – Estimation of distributions of the consumption on the left, Distribution of the ΔSoC on the right for segment 9 for an isosoc constraint

On Fig.3.16, the distributions of the ΔSoC realized by the two methods are shown. We observe that the isosoc constraint is well satisfied by both methods. Indeed computing the mean ΔSoC of both methods give us an mean below 0.05%. The 5th centile and the 95th centile have been computed, see Table 3.2, and have values close to -0.5 and 0.5 that correspond approximately to one second of maximum recharging or discharging.

Δ_{SoC}	median	5-centile	25-centile	75-centile	95-centile
Deterministic	0.0103	-0.5090	-0.0976	0.1234	0.5060
Stochastic	0.0445	-0.4033	-0.0967	0.1272	0.6600

Table 3.2 – SDP vs DDP. Distribution of actual Δ_{SoC} for the iso-SoC case.

3.5 Conclusion

In this chapter, we have proposed a new stochastic method to minimize the consumption of hybrid electric vehicles under known traffic conditions. We have discussed the dependence of the value function with respect to the traffic conditions and battery SoC constraints. The optimal policies resulting have been studied on several speed profiles extracted from actual traffic data. A comparison with the optimal policy from a deterministic optimization with full knowledge of the speed profile has also been presented. The overall consumption from the stochastic method is close to the lower bound given by the deterministic method.

We point out that this stochastic method, using only statistical traffic representation, can be run off-line in order to obtain the optimal policies corresponding to these traffic conditions and SoC constraints. These policies can then be stored and used in an on-line optimization over a whole travel, as we will discuss in the next chapters.

CHAPTER 4

An Optimal State of charge trajectory on a travel with traffic conditions

An article [55], based on this chapter and chapter 3, has been submitted to IEEE Transactions on Control Systems and Technology in 2019.

Contents

This chapter presents the second part of the EMS proposed in chapter 3, the SoC Profile Optimization, which determines the optimal SoC profile that minimizes the consumption on a specific travel. As explained in section 3.1, stochastic dynamic programming methods tends to be unsuitable for real-time implementation due to their high computational cost. Therefore the bi-level method uses the results of the Torque-Split Optimization in order to solve a deterministic dynamic problem, which is fast enough for on-line use.

4.1 Segment Decomposition

We introduce the notion of travel, a travel l is defined by a sequence of contiguous road segments $s \in \mathcal{S}$ of length L_s, which are portions of road with homogeneous topographical characteristics. We call predicted travel when for each segment s of travel l a traffic prediction μ_s, probability law on (V, A) space, has been proposed. We denote μ_l the traffic law along the travel:

$$\mu_l(d) = \sum_{s \in \mathcal{S}} \mu_s 1_{\sum_{i<s} L_i < d \leq \sum_{i \leq s} L_i} \tag{4.1.0.1}$$

4.1.1 Scenarios Decomposition

As seen in the previous chapter, each possible speed profile, i.e. speed according time, can be represented by a scenario ω where $\omega^k = (v^k; a^k)$ represents respectively the mean speed and the mean acceleration of the vehicle over time interval $(t^k; t^{k+1})$. Given a certain speed profile of the vehicle along the travel, we extend the notion of scenario.

The road segments are continuous, thus it is possible to decompose the travel scenario ω into multiple segment scenario as :

$$\omega = (\omega_1, ..., \omega_S) \tag{4.1.1.1}$$

where ω_s is the scenario on the segment s , and belongs to Ω_s the set of possible scenarios on segment s. We define $\Omega := \prod_{s \in [0,S]} \Omega_s$. For a given scenario ω, we note $T_m(\omega)$ the associated decision policy (in the hazard decision framework, the optimal motor torque depends on the operating point), and define the associated cost thanks to the decomposition of the scenario:

$$c(\omega, T_m(\omega)) := \sum_{s \in \mathcal{S}} c(\omega_s, T_m(\omega_s)) \tag{4.1.1.2}$$

Assuming that there exists a probability law on the scenario space Ω, it can be endowed with a Markov chain structure. Each segments scenario space is assuming having a probability law, therefore the transition probabilities between the segment subscenarios are noted :

$$p^k(\omega, \omega') := \mathbb{P}\left[\omega_{s+1} = \omega' | \omega_s = \omega\right] \tag{4.1.1.3}$$

and are computed according to:

$$p^k(\omega, \omega') := \frac{\mathbb{P}\left[\omega_{s+1} = \omega' \text{ and } \omega_s = \omega\right]}{\mathbb{P}\left[\omega_s = \omega\right]} \tag{4.1.1.4}$$

Conversely, given probability transitions (such that the sum of probability transitions starting from a given node is equal to 1), the probability of a scenario is the product of the corresponding sequence of probability transitions. In the adopted memoryless probability transition model, the events $\omega_{s+1} = \omega'$ and $\omega_s = \omega$ are independent, therefore

$$p^k(\omega, \omega') := \mathbb{P}\left[\omega_{s+1} = \omega'\right] \tag{4.1.1.5}$$

The corresponding cost expectation over the whole set of random events Ω is rewritten as

$$\mathcal{C}(T_m) = \sum_{s \in \mathcal{S}} \sum_{\omega_s \in \Omega_s} \mathbb{P}(\omega_s) c(\omega_s, T_m(\omega_s)) \tag{4.1.1.6}$$

4.1.2 Exit times

The entry time ti_s of the vehicle on a segment s is function of the past events contained in the scenario ω. Therefore $\{ti_s\}_s$ is adapted random variables which represents the entry times on segments.

For each segment, the time intervals (ti_s, ti_{s+1}) can be reparametrized to $(0, tf_s := ti_{s+1} - ti_s)$.

Since the traffic conditions are memoryless, the duration tf_s is independent of ti_s. Therefore a local time can be used on each segment s.

4.2 Bi-level decomposition

4.2.1 Global minimal expected consumption problem

Consider the problem of minimizing the total engine consumption over a given predicted travel l. Denote by L_F the total length of the travel l , and by t_F the corresponding final time, which is a random time in view of the previous model of the traffic.The consumption is a random variable as well. The state variables are the state of charge SoC and distance D, and the control is the motor torque T_m, assumed constant over each time interval (t_k, t_{k+1}). So, the problem of *global minimal expected consumption* reads as $(\mathcal{GP})$:

$$\min_{T_m} \quad \mathbb{E}\left[\int_{t=0}^{t_F} C(\mathbf{V}(t), \mathbf{A}(t), T_m(t))\mathrm{d}t + P_F(SoC(t_F))\right] \tag{4.2.1.1}$$

$$s.t\ \forall t, \quad \dot{SoC}(t) = \frac{1}{C_{max}} P_m(\mathbf{V}(t), \mathbf{A}(t), T_m(t)) \tag{4.2.1.2}$$

$$\dot{D}(t) = \mathbf{V}(t) \tag{4.2.1.3}$$

$$t_F = \min\{t, D(t) > L_F\} \tag{4.2.1.4}$$

$$SoC(0) = SoC_0,\ D(0) = 0 \tag{4.2.1.5}$$

$$s.t\ \forall t, \quad T_m(t) \in [T_{min}, T_{max}] \tag{4.2.1.6}$$

$$SoC(t) \in [0, 1] \tag{4.2.1.7}$$

In this Hazard-Decision framework, the control variable depends on the present value of the random variables for speed and acceleration $(\mathbf{V}(t), \mathbf{A}(t))$. Finally, the final cost

$P_F : \mathbb{R} \to \mathbb{R}$ expresses the preference for given final SoC values, typically in order to avoid solutions that would systematically discharge the battery. P_F should be non increasing and we assume it to be continuous.

In the sequel we assume that whatever is the initial state of charge $SoC_0 \in [0,1]$, there exists a sequence of control variables such that the state of charge remains in $[0,1]$. In particular the set below of 'one step' feasible controls

$$\mathcal{T}(SoC, D, \mathbf{V}, \mathbf{A}) := \{T_m \in [T_{min}, T_{max}]; SoC + \frac{h(D,\mathbf{V})}{C_{max}} P_m(\mathbf{V}, \mathbf{A}, T_m) \in [0,1]\} \tag{4.2.1.8}$$

is nonempty, for all $SoC \in [0,1]$ and $(\mathbf{V}, \mathbf{A})$ in the support of the traffic distribution μ_s.

4.2.2 Bi-level formulation

The above global problem $(\mathcal{GP})$ can be solved by dynamic programming techniques. However, this is not adapted to a real-time setting where the traffic could be perturbed on a particular segment by, say, an accident.

For $t \in (t_0; t_F)$, k_t denotes the index of the interval such that $t \in (t^{k_t}; t^{k_t+1})$. Similarly, s_t denotes the segment at time t. Lets rewrite the problem $(\mathcal{GP})$ according to the segment decomposition introduce in section 4.1 as :

$$\min_{T_m} \quad \sum_{s\in\mathcal{S}} \sum_{\omega_s \in \Omega_s} \mathbb{P}(\omega_s) c(\omega_s, T_m(\omega_s)) + P_F(SoC(t_F)) \tag{4.2.2.1}$$

$$s.c\ \forall t, \quad \dot{SoC}(t) = \frac{1}{C_{max}} P_m(\omega^{k_t}, T_m(\omega^{k_t})) \tag{4.2.2.2}$$

$$\dot{D}(t) = V(t) \tag{4.2.2.3}$$

$$SoC(0) = SoC_{init},\ D(0) = 0 \tag{4.2.2.4}$$

$$s.c\ \forall t, \quad T_m(t) \in [T_{min}, T_{max}] \tag{4.2.2.5}$$

$$SoC(t) \in [0,1] \tag{4.2.2.6}$$

We now present a bi-level formulation of this problem, introducing a new decision variable SoC_s^r that will allow us to set a *reference* SoC to be reached at the end of road segment s.

The information structure for the bi-level problem is as follows: the variable T_m is decomposed in variable T_m^s which are original the torque over the segment s. These new variables are adapted to the filtration $\mathcal{F}_s^k$ induced by the subscenario ω_s. By definition, the variable SoC^r is adapted to the filtration $\mathcal{G}_s$ which is the algebra generated by functions of $(SoC(ti_0), ..., SoC(ti_s))$.

$$\min_{SoC^r} \quad \sum_{s\in\mathcal{S}} \mathbb{E}\left[\min_{T_m^s} \mathbb{E}\left[c(\omega_s, T_m(\omega_s))|\, \mathcal{F}_s^k\right] + P_F(SoC(t_F))|\mathcal{G}_s\right] \tag{4.2.2.7}$$

$$s.c\ \forall t, \quad \dot{SoC}(t) = \frac{1}{C_{max}} P_m(\omega^{k_t}, T_m^{s_t}(\omega^{k_t})) \tag{4.2.2.8}$$

$$\dot{D}(t) = V(t) \tag{4.2.2.9}$$

$$SoC(ti_{s+1}) = SoC_s^r \tag{4.2.2.10}$$

$$SoC(0) = SoC_{init},\ D(0) = 0 \tag{4.2.2.11}$$

$$s.c\ \forall t, \quad T_m(t) \in [T_{min}, T_{max}] \tag{4.2.2.12}$$

$$SoC(t) \in [0, 1] \tag{4.2.2.13}$$

Equality constraints, such as 4.2.2.10, are difficult to handle in stochastic optimization due to various scenario that can be infeasible. Therefore the constraints relaxed thanks to a penalization P, that express the preference for the reference SoC. Lets us state the resulting problem ($\mathcal{P}$) :

$$\min_{SoC^r} \quad \sum_{s\in\mathcal{S}} \mathbb{E}\left[\min_{T_m^s} \mathbb{E}\left[c(\omega_s, T_m(\omega_s)) + P(SoC_s^r, SoC(ti_{s+1}))|\mathcal{F}_s^k\right]|\mathcal{G}_s\right] \tag{4.2.2.14}$$

$$s.c\ \forall t, \quad \dot{SoC}(t) = \frac{1}{C_{max}} P_m(\omega^{k_t}, T_m^{s_t}(\omega^{k_t})) \tag{4.2.2.15}$$

$$\dot{D}(t) = V(t) \tag{4.2.2.16}$$

$$SoC(0) = SoC_{init},\ D(0) = 0 \tag{4.2.2.17}$$

$$s.c\ \forall t, \quad T_m(t) \in [T_{min}, T_{max}] \tag{4.2.2.18}$$

$$SoC(t) \in [0, 1] \tag{4.2.2.19}$$

Note that the problem ($\mathcal{P}$) actually encompasses two optimization problem. At the lower problem, the so-called 'micro' problem determines the optimal control on a specific segment s, for a given target SoC_s^r. At the upper level, the 'macro' problem computes the reference SoC trajectory for the travel along the sequence of road segments. We now detail these two problems.

4.2.3 The Micro Problem : An Expected Consumption Model

In the chosen framework of Hazard Decision, the optimal torque distribution depends of the actual speed and acceleration of the vehicle. Therefore the traffic conditions influence the consumption of fuel in a hybrid electrical vehicle. In the previous chapter 2 and 3, it was shown how to use probability distributions to model the traffic conditions and compute the optimal torque. In addition to the torque, the optimization also yields a value function that corresponds to the expected mean consumption plus a penalization term taking into account the final state of charge.

In the bi-level formulation, the above problem corresponds to the 'micro' optimization. Indeed, solving the 'micro' problem for different values of the final state of charge will build a map of expected consumption function $\mathcal{C}$ as function of the initial and final state of charge of the battery and traffic conditions.

$$\mathcal{C}(SoC_i, SoC_f, \mu_s) = \min_{T_m} \quad \mathbb{E}_{\mu_s}\left[c(\omega, T_m(\omega)) + P(SoCf, SoC_{t_f})\right] \tag{4.2.3.1}$$

$$s.c\ \forall t, \quad \dot{SoC}(t) = \frac{1}{C_{max}} P_m(\omega^{k_t}, T_m(t, \omega^{k_t})) \tag{4.2.3.2}$$

$$\dot{D}(t) = V(t) \tag{4.2.3.3}$$

$$SoC(0) = SoC_i,\ D(0) = 0 \tag{4.2.3.4}$$

$$s.c\ \forall t, \quad T_m(t) \in [T_{min}, T_{max}] \tag{4.2.3.5}$$

$$SoC(t) \in [0, 1] \tag{4.2.3.6}$$

All these 'micro' optimizations can be perfomed once for all off-line and stored for future use. Furthermore, the computation can be parrallelized according to the traffic conditions μ, and the final state of charge constraint SoC_f.

Remark on computation of the micro problem

Note in Fig.4.1 , that the hierarchy between the four cost maps is consistent with the traffic conditions for the clusters (see also Fig. 3.5,3.6,3.7,3.8): cluster 1 has the higher speed and relies more heavily on the engine, hence it has a higher cost overall; cluster 2 has the lowest speed and also the lowest cost; clusters 3 and 4 correspond to intermediate traffic conditions.

Remark: the cost maps appear symmetric w.r.t the diagonal, which is not surprising: since the actual value of the SoC does not appear in the equations of the micro problem, these problems are invariant by translation on (SoC_0, SoC_f). This can of course be used to significantly reduce the computational cost of obtaining these maps. The exception is when the constraint $SoC \in [0, 1]$ becomes active, therefore all micro problems with boundary conditions close to 0 or 1 have to be solved explicitly.

4.2.4 Macro problem

Problem statement

Assuming that the cost map $\mathcal{C}$ is already computed from the micro problems, we introduce the following 'macro' problem, a variant of the problem $\mathcal{P}$, in which the decision is the deterministic trajectory of reference SoC^r at the end of each road segment:

$$\min_{SoC^r} \quad \sum_{s=0}^{S-1} \mathcal{C}(SoC_s^r, SoC_{s+1}^r, \mu_s) \tag{4.2.4.1}$$

$$s.t.\ \forall s, \quad SoC_s^r \in [0, 1] \tag{4.2.4.2}$$

$$SoC_0^r = SoC_{intial} \tag{4.2.4.3}$$

$$SoC_S^r = SoC_{final} \tag{4.2.4.4}$$

This "macro' problem will perform the optimization of the state of charge trajectory along a complete travel, using the expected consumption function computed off-line.

Note that the problem is deterministic since the conditions of traffic are assumed to be time-independent.

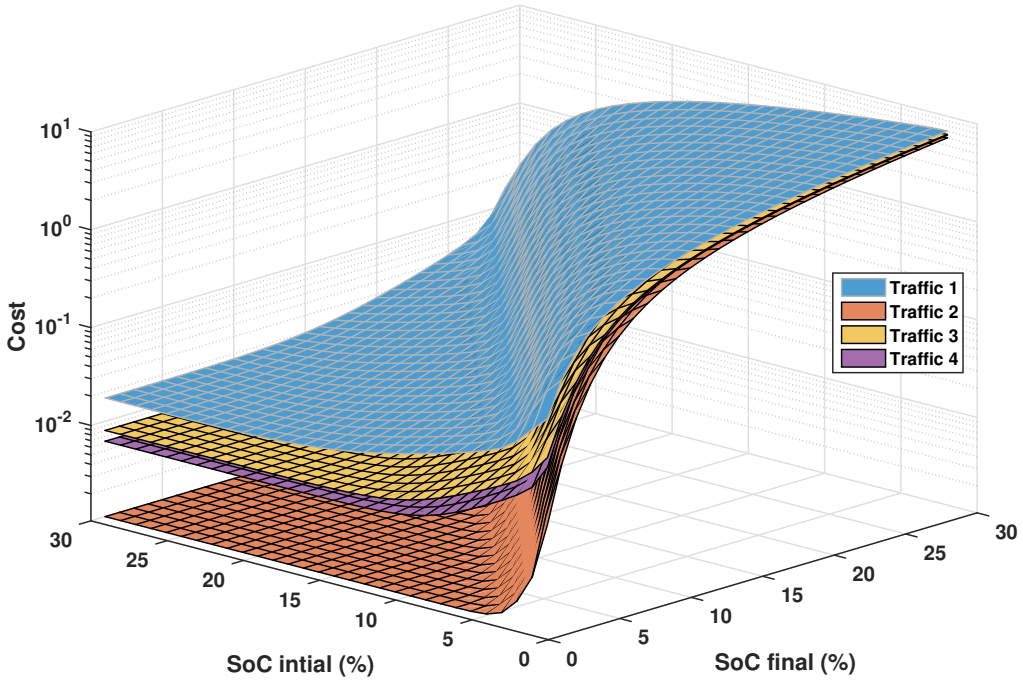


Figure 4.1 – Illustrative cost map v_s for a family of micro problems.

Dynamic Principle

We now formulate the Dynamic Programming Principle in order to solve the macro problem. The value function denoted U_{macro}, and satisfies the following equation:

$$U_{macro}(s, SoC) = \inf_{SoC^r} \left(\mathcal{C}(SoC, SoC^r, \mu_s) + U_{macro}(s+1, SoC^r)\right) \tag{4.2.4.5}$$

$$U_{macro}(S, \cdot) = P(\cdot) \tag{4.2.4.6}$$

Dynamic Programming Algorithm

We recall here the classical algorithm to compute the value function.

Algorithm 3: DDP Algorithm

$s = S$
for $s > 0$ **do**
  for $SoC \in \mathbb{X}_s$ **do**
    $U_{macro}(s, SoC) = \infty$;
    for $SoC^r \in \mathbb{U}_s$ **do**
      $U_{macro}(s, SoC)) = \min(U_{macro}(s, SoC)), \min_{SoC^r}(\mathcal{C}(SoC, SoC^r, \mu_s)$
        $+U_{macro}(s+1, SoC^r)))$;
  $s = s - 1$;

Control of the vehicle

As presented above, the variables SoC_s^r are targets to reach at the end of the segments. The control obtained by the penalized micro problem tries to reach this target, but due to the penalization, the target is not reached exactly. In order to simulate the behaviour of a vehicle using the bi-level optimization, we propose in Alg.4 an open-loop control algorithm.

Algorithm 4: Open Loop Simulation Algorithm

SoC^r Selection from 'macro' value function
$SoC_0^r = SoC_{init}$
while $s < S$ **do**
  $s = s+1$; $SoC_s^r = \text{argmin}_{SoC_s^r}(\mathcal{C}(\mathbf{SoC_{s-1}^r}, SoC_s^r, \mu_s) + \tilde{U}_{macro}(s+1, SoC_s^r))$;

Torque Selection from 'micro' value function
$SoC^0 = SoC_{init}$;
$k = 0$;
while $s < S$ **do**
  $D^k = 0$;
  while $D^k < L_s$ **do**
    $T_m^k = \text{argmin}_{T_m}(c(SoC^k, D^k, T_m, \omega^k) + \tilde{V}_{s,\Delta SoC_s,\mu}(f(SoC^k, D^k, T_m, \omega^k)))$;
    $(SoC^{k+1}, D^{k+1}) = f(SoC^k, D^k, T_m^k, \omega^k)$;
    $k = k+1$;
  $s = s+1$;

If the micro simulation is fast enough, the closed-loop variant in Alg.5 may be used instead. The difference with the open-loop algorithm is that the SoC_s^r is updating with the actual final SoC from the 'micro' optimization.

Algorithm 5: Closed Loop Simulation Algorithm

SoC^r Selection from 'macro' value function
$SoC^0 = SoC_{init}$;
$k = 0$;
while $s < S$ **do**
  $D^k = 0$;
  $s = s+1$;
  $SoC_s^r = \text{argmin}_{SoC_s^r}(\mathcal{C}(\mathbf{SoC^k}, SoC_s^r, \mu_s) + \tilde{U}_{macro}(s+1, SoC_s^r))$;
  Torque Selection from 'micro' value function
  while $D^k < L_s$ **do**
    $T_m^k = \text{argmin}_{T_m}(c(SoC^k, D^k, T_m, \omega^k) + \tilde{V}_{s,\Delta SoC_s,\mu}(f(SoC^k, D^k, T_m, \omega^k)))$;
    $(SoC^{k+1}, D^{k+1}) = f(SoC^k, D^k, T_m^k, \omega^k)$;
    $k = k+1$;

4.3 Numerical results

Simulations have been run with the vehicle proposed in section 3.2.1 on a portion of the highway A7 near Lyon. The travel, see Fig. 4.2, is composed by 26 different segments. The method proposed in chapter 2 has been used on data collected thanks

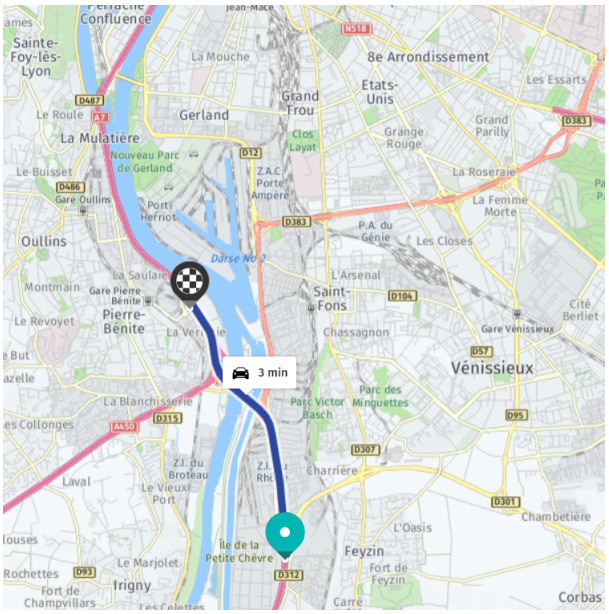


Figure 4.2 – Travel on A7 used in simulations.

to GecoAir. Then the functions $\mathcal{C}(\cdot,\cdot,\mu_s)$ have been computed thanks to the method proposed in chapter 3 on a SoC discrete grid with a step of 1%.

Optimal policy for macro problem

We show in Fig. 4.3 a sample optimal trajectory for the macro problem, for which we also recomputed the sequence of micro problems along the travel. Top graph displays a real speed profile recorded thanks to 'Geco air'. Middle graph shows the optimal policy for the macro problem (i.e SoC^r), as well as the actual SoC corresponding to the sequence of micro problems. Bottom graph indicates the engine and motor torques of the vehicle recovered thanks to the value function of the micro problems. Finally, we also optimize the whole travel using DDP, taking the speed profile from the micro problems. The corresponding SoC trajectory is also shown on the middle graph.

For this simulation we set the conditions $SoC_0 = 0.25, SoC_f = 0.24$. We observe that both the macro and DDP solutions satisfy the final SoC correctly, although the SoC trajectories are different. The consumption is 0.2363 for bi-level and 0.1991 for DDP. The over-consumption of 18% for the bi-level method seems reasonable against the deterministic solution with full knowledge of the speed profile. The SoC trajectory from the micro problems initially diverges a bit from the macro target, but catches up later.

As with the micro problems, we now compare the bi-level method with the deterministic solution over a thousand known speed profiles. The distributions of the consumption and $\Delta_{SoC} := SoC(t_f) - SoC(t_0)$ are shown on Fig. 4.4. The average consumption

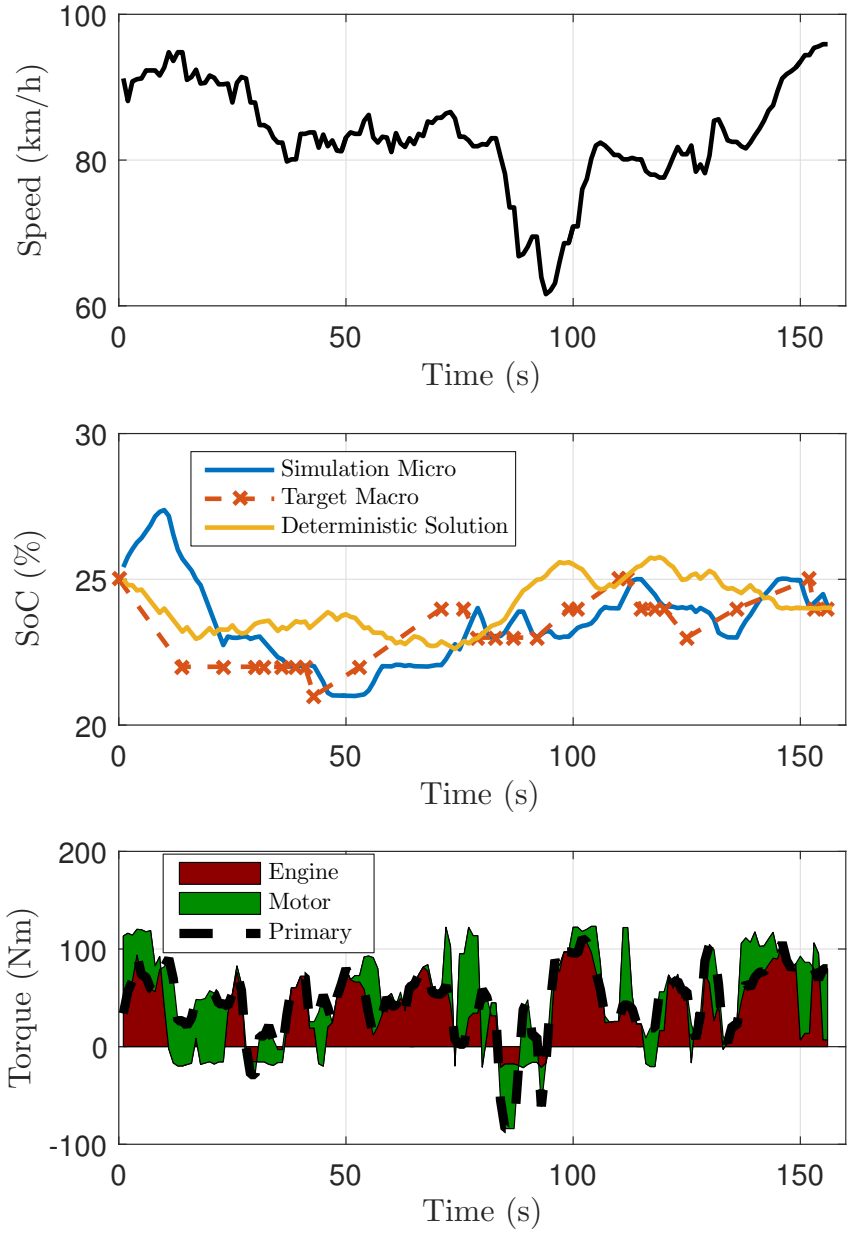


Figure 4.3 – Macro problem: optimal policy. Bi-level optimization vs DDP. Conditions $SoC_0 = 0.25, SoC_f = 0.24$.

obtained by the bi-level method is $6.58L/100km$, and the deterministic optimization obtained an average consumption of $5.70L/100km$. The average consumption is 15% higher for the stochastic bi-level approach, which seems reasonable considering the relatively large step size used for the discretization of SoC^r. Also, the bi-level Δ_{SoC} is centered around -1%, as expected, even if less strongly than for the deterministic approach, see Table 4.1.

Overall, the bi-level method seems to give satisfying results compared to a method with full traffic information.

Δ_{SoC}	median	5-centile	25-centile	75-centile	95-centile
Deterministic	-0.9726	-1.7242	-0.9899	-0.9442	-0.2915
Bi-Level	-1.0339	-6.3671	-1.3986	-0.7537	0.1881

Table 4.1 – Macro problem: Bi-Level vs DDP. Distribution of actual Δ_{SoC} for depleting 1% of the battery.

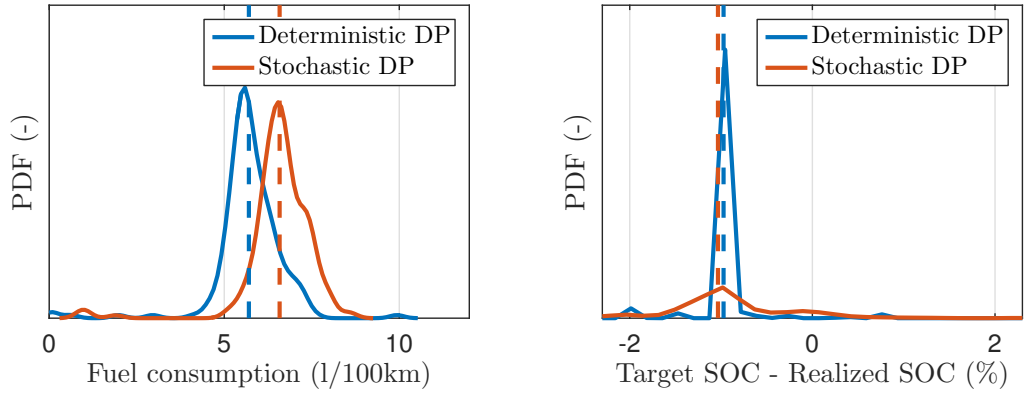


Figure 4.4 – Macro problem: Consumption distribution over 939 speed profiles. Bi-level optimization vs DDP. Conditions $SoC_0 = 0.25, SoC_f = 0.24$.

CPU times

The simulation times are recapped in Table 4.2. For the off-line micro level, one SDP problem gives the cost map for all possible initial SoCs, so we only need to solve the family of problems for all possible final SoCs. Furthermore, all the micro problems are independent. On the other hand, the CPU time for the macro problem is quite small, as required for an on-line implementation. The key element for this fast macro problem is the handling of the stochastic traffic conditions at the micro level, leaving only a simple deterministic optimization at the macro level. For comparison purposes, we also include the time to optimize a full travel using DDP with complete traffic information, as well as an estimate of the cost for solving the same problem with stochastic traffic conditions.

	CPU time
Micro level (SDP, 1 segment, 1 SoC_f)	$\sim$ 1h
Macro level (DDP, deterministic SoC)	$\sim$ 10s
Reference: DDP over the whole travel	$\sim$ 2h
Estimation of the SDP over the travel	($\sim$ 200h)

Table 4.2 – CPU times. Estimation with a 10×10 speed/acceleration discretization due to the linear complexity of SDP.

4.4 Conclusion

In this chapter, we have presented a bi-level method for the energy management of a hybrid vehicle. More precisely, the aim is to minimize the fuel consumption of the thermal engine over a fixed travel, assuming that the vehicle follows the (stochastic) traffic

conditions of the road. We consider a subdivision of the road network into small segments, for which typical traffic conditions are modeled as probability distributions in the (speed,acceleration) plane. The key point in order to obtain a method fast enough for online use is that the stochastic traffic conditions are completely handled at the micro level, leaving only a simple deterministic optimization at the macro level. This optimization relies on an energy cost map over all road segments, for all initial and final SoC conditions. We compute these costs at the micro level by solving offline a family of stochastic optimization problems under traffic constraints, see chapter 3.

Numerical simulations carried out using actual traffic data from a highway portion near Lyon (France) indicate that the bi-level approach performs in a satisfying manner, with a limited over-consumption compared to a deterministic solution using fully known traffic information.

In the next chapter 5, this bi-level method is extended to the eco-routing problem, by adding the path decision to the variable.

CHAPTER 5

Application to the optimal routing problem under traffic conditions

An article [57], based on this chapter, has been submitted to the 21st IFAC World Congress.

Contents

5.1 Introduction

Road networks usually allow several paths to reach a destination from a given starting position. While path has traditionnally been choosen to minimize travel time, methods to minimize the fuel consumption have received increased attention in the recent years. This new criterion gives rise to the so-called eco-routing problem. The fuel savings between the optimized through eco-routing and the path naturally chosen by the drivers can be important, up to 25% [2, 32]. Besides, distance minimal paths have been shown to differ from the eco-routing path, especially in congested traffic [6]. This indicates that eco-routing planning needs to take into account the traffic conditions.

Most methods propose to solve the eco-routing planning based on shortest path algorithms, using weighted graphs to represent the road network with edge costs corresponding to the consumption, see [23]. Dijkstra algorithm [25] or A^* algorithm [43] can be used when edge costs are positive, or Ford-Bellman algorithm [7] when edge costs can be negative. In order to determine these edge costs, an estimator is needed. Consumption estimators, that take traffic conditions into account, are usually divided in two main categories: macroscopic models based on closed algebraic forms [59], and microscopic models based on differential equations [74]. We refer to section 2.1 for a more extended description of traffic models.

In the case of HEVs, the vehicle can use the electric motor to reduce the fuel consumption or recover energy thanks to regenerative braking. Therefore if costs are expressed in terms of energy, negative costs can appear. However, path-searching algorithms for negative costs tends to have a high computational cost. Thus it appears more efficient to express costs in terms of fuel consumption, which requires knowledge of the torque policy of the vehicle. We adapt the bi-level approach introduced in chapters 3 and 4, in order to compute an optimal path adapted to the optimal torque policy of the vehicle following the traffic conditions.

5.2 Modeling the issue

5.2.1 A road graph with traffic conditions

A road graph

The consumption of a vehicle is influenced by the road profile, namely the allowed speed, the slope, and other parameters such as weather conditions. It seems natural to choose a road network model that includes the time independent parameters. A classical way is to use a graph where roads are the edges and intersections the nodes. The characteristics of the road portions are therefore the attributes of the edges. Figure 5.1 represents the road network of Paris.

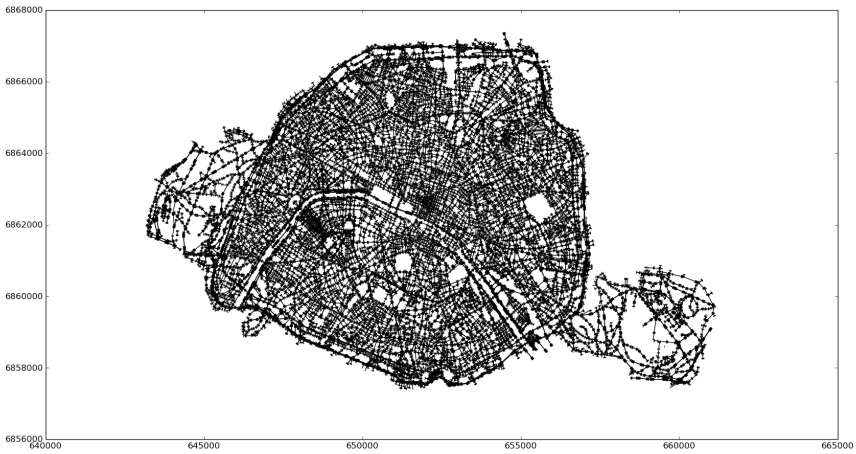


Figure 5.1 – Road Network of Paris

As previously mentioned, traffic conditions have a large influence on the consumption of the vehicles. Therefore it is desirable to include the traffic information in the attributes of the edges. Such a traffic model was introduced in chapter 2 and is recalled in the following.

Probabilistic Traffic Model

We recall that we consider a decomposition into small *segments* based on topographic aspects. For a given vehicle, we denote by t_s its entry time on the segment s and define the related time grid $t_{s,k} := t_s + kh_0$, for $k \in \mathbb{N}$ and the time step h_0, which is set in simulations at $h_0 = 1s$, consistent with the typical reaction time of the combination driver/vehicle. We assume that the speed and acceleration of the vehicle are random variables $(\mathbf{V}(t), \mathbf{A}(t))$ that are i.i.d constant over each *grid interval* $(t_{s,k}, t_{s,k+1})$. The corresponding *traffic distribution* μ^s on segment s has finite support $\mathrm{supp}(\mu^s)$ included in $\mathbb{R}_+ \times \mathbb{R}$, see Fig.5.2.

The assumption that the traffic distributions do not depend on time is an important simplification, allowing to reduce considerably the burden of computations (since otherwise we would need an additional state variable representing the time, and handle distribution changes at times other than entry times, etc).

The central assumption, that the vehicle speed and acceleration coincide with the random variables $(\mathbf{V}(t), \mathbf{A}(t))$, can be interpreted as the vehicle 'following the traffic'. In [56] we showed how to construct such a statistical traffic model and how to derive estimations for the energy consumption of a given car. In particular, the data size can be reduced by clustering techniques applied to the traffic distributions. Even with a small number of clusters an accurate prediction of the energy consumed is obtained.

5.2.2 A weighted state graph for hybrid electrical vehicle

The state graph

In the case of ICE vehicles, the optimal path, i.e. the successive positions of the vehicle on the road graph, is sufficient to define the optimal strategy. In the case of a hy-

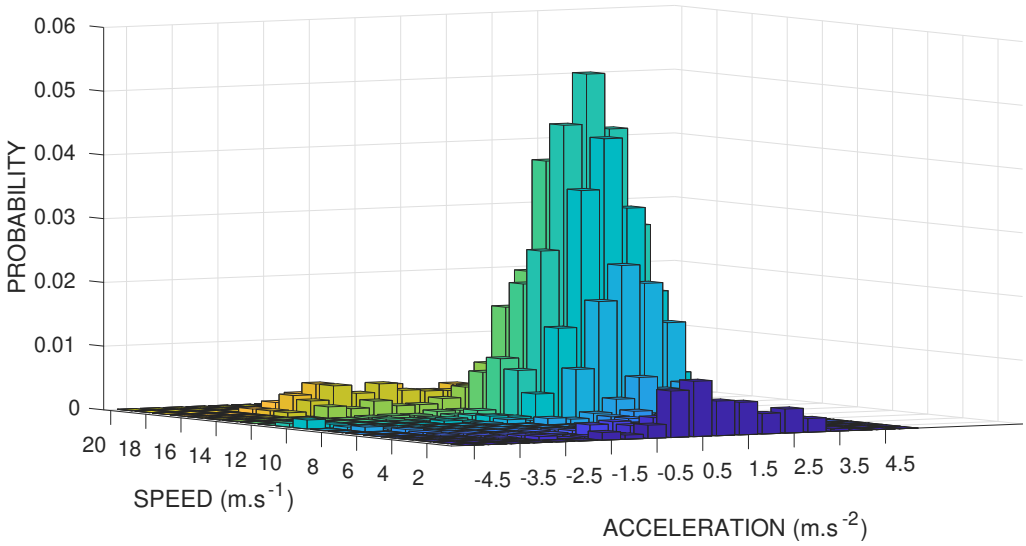


Figure 5.2 – Distribution

brid vehicle, the energy stored in the battery can be used to reduce the fuel consumption, adding a supplementary state, the state of charge (SoC) of the battery. Therefore the optimal consumption policy consists in the successive vehicle positions and states of charge of the battery.

Using the lexicographic product between graphs, see Fig.5.3, we define the 'state graph', denoted Γ, whose nodes have a form (N, SoC) with N a node of the road graph and SoC in a non empty discrete set of state of charge values. Therefore an edge between (N_1,SoC_1) and (N_2,SoC_2) exists only if there is an edge between N_1 and N_2.

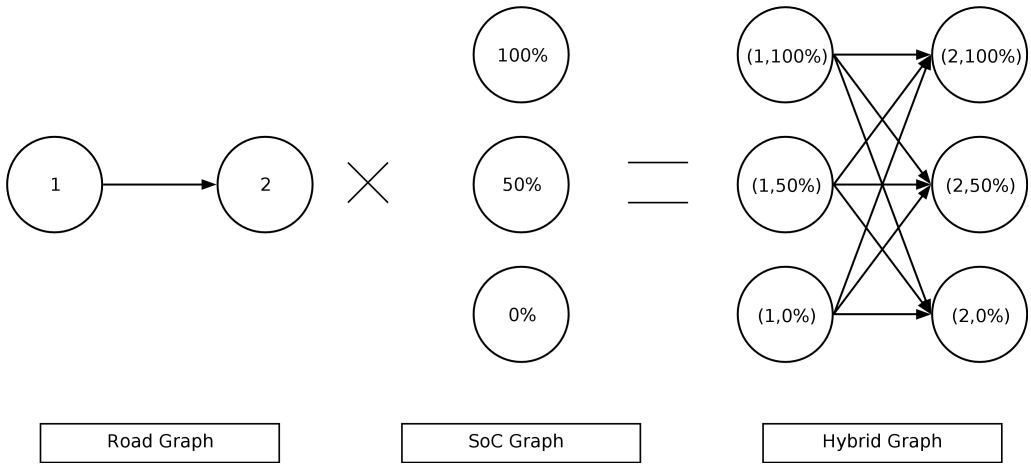


Figure 5.3 – Explicative Schema of lexicographic product

Finding the optimal path on the 'state graph' provides an approximation of the solution of the eco-routing problem for HEV. The degrees of freedom given by the electric motor of the HEV can be used to optimize the consumption, for instance if it can be better to spend more fuel on an edge to recharge the battery.

Cost of an edge

Since the consumption of the vehicle depends on the road and the traffic conditions, each edge has a specific expected cost. In order to obtain the edge cost between two nodes (N, SoC) and (N', SoC'), i.e. the expected consumption on the road segment $s_{NN'}$, the method proposed in [55] is used.

We recall the formulation of the 'micro' problem defined in chapter 3 for the road segment corresponding to the edge of the graph. The resulting problem $(\mathcal{P}^{s_{NN'}}_{micro})$ is

$$\min_{T_m} \mathbb{E}\left[\int_{t=0}^{t_f} C(\mathbf{V}(t), \mathbf{A}(t), T_m(t))\mathrm{d}t + P_{s_{NN'}}(SoC_{s_{NN'}}(t_f), SoC')\right] \tag{5.2.2.1}$$

$$s.t\ \forall t,\ \dot{SoC}_{s_{NN'}}(t) = \frac{1}{C_{max}} P_m(\mathbf{V}(t), \mathbf{A}(t), T_m(t)) \tag{5.2.2.2}$$

$$\dot{D}_{s_{NN'}}(t) = \mathbf{V}(t) \tag{5.2.2.3}$$

$$T_m(t) \in [T_{min}, T_{max}] \tag{5.2.2.4}$$

$$SoC_{s_{NN'}}(t) \in [0, 1] \tag{5.2.2.5}$$

$$SoC_{s_{NN'}}(0) = SoC,\ D_{s_{NN'}}(0) = 0 \tag{5.2.2.6}$$

$$t_f = \min\{t, D_{s_{NN'}}(t) > L_{s_{NN'}}\} \tag{5.2.2.7}$$

Alternatively we can reformulate the above problem in a purely discrete time setting, dropping indexes s for the state variables:

$$\min_{T_m} \mathbb{E}\left[\sum_{k=0}^{k_f} h^k C(\mathbf{V}^k, \mathbf{A}^k, T^k_m,) + P_{s_{NN'}}(SoC^{k_f}, SoC')\right] \tag{5.2.2.8}$$

$$s.t\ \forall k,\ SoC^{k+1} = SoC^k + \frac{h^k}{C_{max}} P_m(\mathbf{V}^k, \mathbf{A}^k, T^k_m) \tag{5.2.2.9}$$

$$D^{k+1} = D^k + h^k \mathbf{V}^k \tag{5.2.2.10}$$

$$T^k_m \in [T_{min}, T_{max}] \tag{5.2.2.11}$$

$$SoC^k \in [0, 1] \tag{5.2.2.12}$$

$$h^k := \min(h_0, (L_{s_{NN'}} - D^k)/\mathbf{V}^k) \tag{5.2.2.13}$$

$$k_f := 1 + \max\{k, D^k + h_0 \mathbf{V}^k < L_{s_{NN'}}\} \tag{5.2.2.14}$$

$$SoC^0 = SoC;\ D^0 = 0. \tag{5.2.2.15}$$

The expected consumption, denoted $\nu_{s_{NN'}}(SoC, SoC')$, on a road segment $s_{NN'}$ with known traffic conditions is the value of the discrete version of problem $(\mathcal{P}^{s_{NN'}}_{micro})$.

5.2.3 Optimal path for HEV under traffic conditions

It is now possible to formulate an eco-routing problem for HEVs as finding the optimal path on the 'state graph' Γ between a start node S and a destination node D. We denote by Γ_{SD} the set of paths between S and D in the 'state graph'. The same idea as in chapter 4 is used: this optimal path problem consists in finding a target state of charge trajectory SoC^r. We introduce a_i (b_i) the maximum charge (respectively discharge) to reduce the

search space, with $a_i = 1$ and $b_i = 1$, corresponding to the original problem. The choice of a_i (b_i) can be guided by the vehicle and segment characteristics. In the sequel, we assume:

$$a_i > 0, b_i > 0 \tag{5.2.3.1}$$

$$\begin{aligned}
&\underset{\gamma \in \Gamma_{SD}}{\text{minimize}} && \sum_{i \in \gamma} \nu_{N_i N_{i+1}}(SoC_i^r, SoC_{i+1}^r) && (5.2.3.2)\\
&\text{s.t} && \forall i \in \gamma, SoC_i^r \in [0,1] && (5.2.3.3)\\
& && \forall i \in \gamma,\ SoC_{i+1}^r - SoC_i^r \in [-a_i, b_i] && (5.2.3.4)
\end{aligned}$$

5.3 Error analysis

As the SoC is discretized, it is useful to analyse the error between the discrete problem (P_h) and the continuous one (P). In order to analyze it, let be γ a path on the road network, and consider the consumption minimization problem on path γ. We note the continuous problem (P^γ), and the discrete problem (P_h^γ). We define the criterion:

$$F^\gamma(SoC^r) := \sum_{i \in \gamma} \nu_{N_i N_{i+1}}(SoC_i^r, SoC_{i+1}^r) \tag{5.3.0.1}$$

$$\begin{aligned}
&\underset{SoC^r}{\text{minimize}} && F^\gamma(SoC^r) && \\
&\text{s.t} && \forall i \in \gamma, SoC_i^r \in [0,1] && (P^\gamma)\\
& && \forall i \in \gamma, SoC_{i+1}^r - SoC_i^r \in [-a_i, b_i] &&
\end{aligned}$$

$$\begin{aligned}
&\underset{SoC^r}{\text{minimize}} && F^\gamma(SoC^r) && \\
&\text{s.t} && \forall i \in \gamma, SoC_i^r \in \{0, h, ..., 1\} && (P_h^\gamma)\\
& && \forall i \in \gamma, SoC_{i+1}^r - SoC_i^r \in [-a_i, b_i] &&
\end{aligned}$$

Denote the value of (P_h^γ) by V_h^γ, and the value of (P^γ) by V^γ. Let the eco-routing problem (P) be

$$\min_\gamma (V^\gamma) \tag{P}$$

and its discrete approximation (P_h) be

$$\min_\gamma (V_h^\gamma). \tag{P_h}$$

Theorem 4. *Discretization Error*

Let the criterion F^γ be Lipschitz continuous with constant L and let S^* be a solution of (P^γ). Then there exists S_h feasible for (P_h^γ), such that $|S_h - S^*| = O(h)$ and

$$F^\gamma(S^*) \leq F^\gamma(S_h) \leq F^\gamma(S^*) + O(h). \tag{5.3.0.2}$$

Proof. Let S^* a admissible solution for (P^γ). Let $\hat{S} = (1-\varepsilon)S^*$.

Then $\hat{S} \in [0,1]$ and also $\hat{S}_{i+1} - \hat{S}_i \in [-(1-\varepsilon)a_i, (1-\varepsilon)b_i]$.

Define the discretized grid $H = \{0, h, ..., 1\}$, and a projector Π_h over H such that

$$\Pi_h(x) \in \underset{h \in H}{\operatorname{argmin}}(|x-h|) \tag{5.3.0.3}$$

Let $\hat{S^h} := \Pi_h(\hat{S})$. Then $\hat{S^h} \in \{0, h, ..., 1\}$ and

$$\hat{S}_{i+1} - \hat{S}_i \in [-(1-\varepsilon)a_i - h, (1-\varepsilon)b_i + h]. \tag{5.3.0.4}$$

Therefore $\hat{S}_h$ is feasible for problem (P_h^γ) if and if only:

$$h - \varepsilon b_i \leq 0 \tag{5.3.0.5}$$

$$\varepsilon a_i - h \geq 0 \tag{5.3.0.6}$$

Thanks to assumption (5.2.3.1), we can take :

$$\varepsilon = \frac{h}{\min(\min_i(a_i), \min_i(b_i))} \tag{5.3.0.7}$$

Therefore, $\varepsilon = O(h)$. An admissible solution of (P_h^γ) is at $O(h)$ distance of an admissible solution of (P^γ). Since the objective function F^γ is Lipschitz continuous, the conclusion follows. □

Corollary 5. *Under hypothesis of theorem 4, we have that*

$$val(P) \leq val(P_h) \leq val(P) + O(h) \tag{5.3.0.8}$$

5.4 Find the shortest path

In the problem (5.2.3.2), all edge costs are non negative since the vehicle consumes fuel and cannot produce it.

5.4.1 Dijkstra algorithm

A common algorithm to find the shortest path between two points in a directed graph where all the edges have positive weights, is the Dijkstra algorithm [17]. The principle of the Dijkstra algorithm is to separate the nodes in two categories, already *explored* nodes and those *unvisited*, while exploring the graph from the source until you find the destina-

tion. The algorithm can be described as Alg.6 where $\triangle$ is the symmetric difference.

Algorithm 6: Dijkstra Algorithm

```
Explored_Nodes={}
Unvisited_Nodes={source}
while Unvisited_Nodes is not empty do
    u= argmin_{u∈Unvisited_Nodes} value(u)
    if u is destination then
        We return path[u]
    else
        Explored_Nodes = Explored_Nodes ∪ {u}
        Unvisited_Nodes = Unvisited_Nodes △ {u}
        forall v neighbour of u do
            New_Value= value(u) + cost(u,v)
            if New_Value < value(v) then
                value(v)=New_Value
                path[v]=path[u] ∪ {u}
            if v ∉ Unvisited_Nodes then
                Unvisited_Nodes = Unvisited_Nodes ∪ {v}
No Path exists
```

5.4.2 The A* algorithm

The A* algorithm is a heuristic algorithm used to determine the optimal routing on a weighted graph described the first time in [43]. To determine the optimal routing in a graph, the A* algorithm uses the principle of the Dijkstra algorithm, but thanks to a heuristic distance h, decreases the number of nodes explored.

By choosing the node with the minimum expected distance each time, we avoid exploring nodes having an expected distance greater than the optimal routing.

The A* algorithm is shown in Alg.7.

The optimality of A* algorithm

The A* algorithm uses a heuristic to limit the number of explored node. We recall the results of [70] that it is possible to insure that the response given by the algorithm A^* is the optimal path under some assumptions over the heuristic distance .

Definition 1. *Admissibility*

A heuristic h is admissible if it never overestimates the real cost to reach the goal. In other terms,

$$\forall v \in \Gamma, h(v) \leq h^*(v) \tag{5.4.2.1}$$

where $h^*(v)$ is the minimum cost between v and the goal node.

Theorem 6. *If the heuristic h is admissible then the A* algorithm stops and finds an optimal path between the source s and the destination d, if it exists.*

Algorithm 7: A* Algorithm

```
Explored_Nodes={}
Unvisited_Nodes={source}
while Unvisited_Nodes is not empty do
    u= argmin_{u∈Unvisited_Nodes} estimated_value(u)
    if u is destination then
        We return path[u]
    else
        Explored_Nodes = Explored_Nodes ∪ {u}
        Unvisited_Nodes = Unvisited_Nodes △ {u}
        forall v neighbour of u do
            New_Value= value(u) + cost(u,v)
            if New_Value < value(v) then
                value(v)=New_Value
                estimated_value(v)=value(v)+heuristic(v)
                path[v]=path[u] ∪ {u}
            if v ∉ Unvisited_Nodes then
                Unvisited_Nodes = Unvisited_Nodes ∪ {v}
No Path exists
```

Complexity of the A* algorithm

In the case of long travel, the road network is a graph with a large number of edges and nodes. Therefore the complexity of the A* algorithm needs to be discussed. We recall the worst case complexity [64] .

In the worst case scenario, the destination node is explored after all the other nodes on the graph, and after that all the edges have been explored too. If n is the number of nodes and m the number of edges, the complexity is $O(n * operations + m * operations)$. The only operations which is not in $O(1)$ is the minimization over the nodes, that can be made in $O(log(n))$. Then the complexity is $O(n * \log(n) + m)$ and we have that $m \leq n^2$. So the complexity of the A* algorithm in the worst case possible is $O(n^2)$.

However, the number of nodes of the state graph depends on the discretization of the SoC^r as well as the number of edges. The number of nodes of the state graph is the product of the number of intersections n in the road network and the number of SoC divisions created. The number of edges of the state becomes m/h^2. Then the complexity of the A* algorithm according to the problem is $O(n/h \log(n/h) + m/h^2$.

5.4.3 Choice of the heuristic

We propose here an admissible heuristic for the case of the hybrid vehicle eco-routing. We define h_c an estimate of the consumption to reach the physical destination

$$h_c := \eta_m \alpha_0 L \tag{5.4.3.1}$$

with L is the travel length, η_m be the maximum efficiency of the ICE to convert fuel to mechanical energy, α_0 defined in section 3.2.1, and m the mass of the vehicle.

Similarly, we define h_{SoC} an estimate of the consumption to reach the desired state of charge

$$h_{SoC} := \eta_e C_{max}(SoC_f - SoC_{current}) \tag{5.4.3.2}$$

with η_e the maximum efficiency of the ICE to convert fuel to battery charge.

Finally, we must take into account the possibility of regenerative braking that can recover the kinetic energy of vehicle. Therefore, we define $h_{kinetic}$

$$h_{kinetic} := \frac{\eta_c m}{2} v_{max} \tag{5.4.3.3}$$

with η_c a conversion factor of the kinetic energy into fuel according to the ICE and v_{max} the maximum speed. Since we want to reach a specific destination with a final state of charge, the heuristic is taken as the sum of these estimates. Additionally, we take into account that the fuel cannot be produced by the vehicle, giving the final expression:

$$h := \max(0, h_c + h_{SoC} + h_{kinetic}) \tag{5.4.3.4}$$

5.5 Numerical simulations

We now present numerical simulations for the eco-routing method. We study a simple road network comprised of a small ring with congested traffic with a mean speed of 40 km/h, enclosed in a larger ring with fluid traffic with a mean speed of 100 km/h. This can be considered as a very simplified model of a typical road network of ring roads network around a city. For the sake of simplicity, each segment of the presented network has the same topological aspect. In particular, we are interested in comparing the solution from the eco-routing, called 'eco-path' in the following, with the fastest path. We first study a specific travel with a fixed origin and destination, and then give some more general results on all possible travels on the network.

5.5.1 Study of a single travel

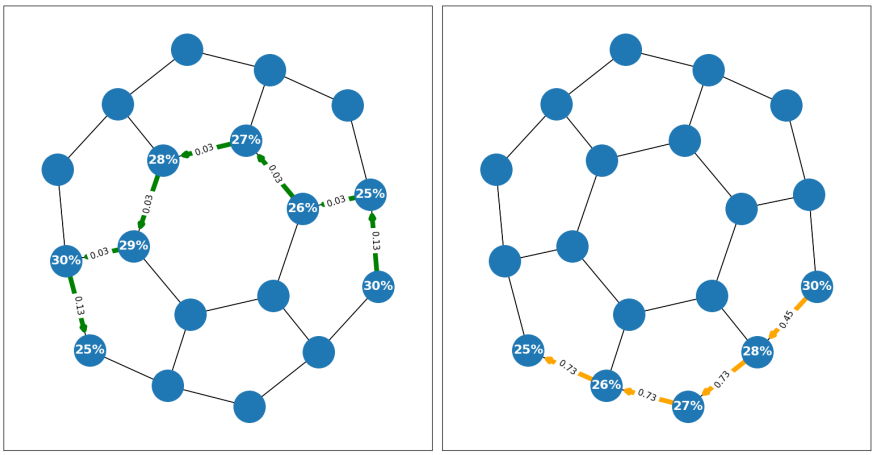


Figure 5.4 – Eco-path (left) and Fastest Path (right) - $(SoC_i = 30\%, SoC_f = 25\%)$

We show on Fig.5.4 the eco-path (green, left graph) and fastest path (orange, right graph) for a given travel with conditions $(SoC_i = 30\%, SoC_f = 25\%)$. Keep in mind that the considered vehicle has a low capacity battery, see C_{max} in Table 3.1, allowing for relatively significant SoC changes even on small road segments. The SoC values at the end of each road segment are indicated on the nodes, while the value from the micro problems are displayed along each edge. In this particular example, the fastest path takes the 4 fast segments on the outer ring, while the eco-path uses the slower inner segments to reduce the consumption, with 7 segments in total.

Path	Value	Cons. (L)	SoC_f (%)	Time (s)	Dist. (km)
Eco	0.41	0.012	21	163	2.14
Fastest	2.66	0.038	4	44	1.22

Table 5.1 – Comparison of Eco and Fastest Path - $(SoC_i = 30\%, SoC_f = 25\%)$.

Table 5.1 summarizes for both paths the sum of values of the micro problems, consumption, final SoC, time and distance for a $SoC_f = 25\%$ constraints. The time is based on the average speed of each segment. The consumption and final SoC are recomputed by taking the average of 1000 resimulated travels using the local optimal policies determined by the eco-path, with i.i.d. sampling according to the traffic conditions for the (v, a) values. For this sample travel the eco-path consumption is one third of the fastest path, for a double distance and four times longer time. Note that the difference in terms of value is greater than the difference in consumption, which indicates that the eco-path has a better chance of following the reference SoC trajectory, since the value function of the micro problems is the sum of the consumption and penalty for the final SoC constraint at the end of the segment. This also shows in the final SoC value, with the eco-path being much closer to the prescribed $SoC_f = 25\%$, reaching 21% while the fastest path ends up at only 4%.

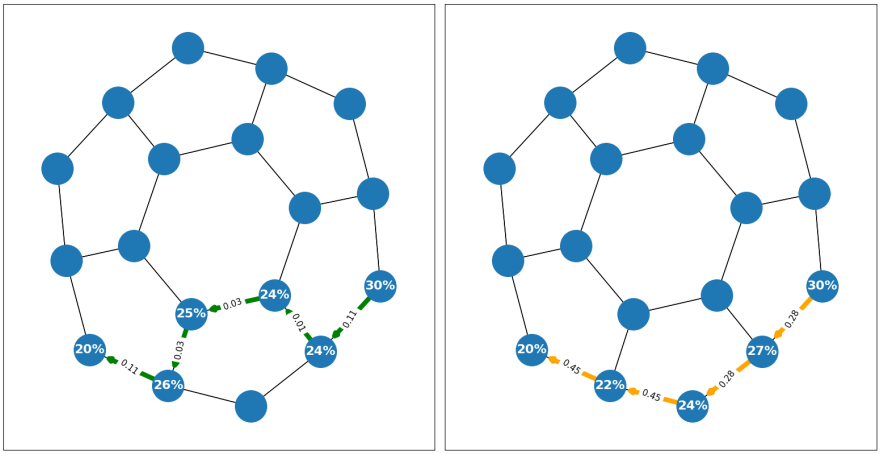


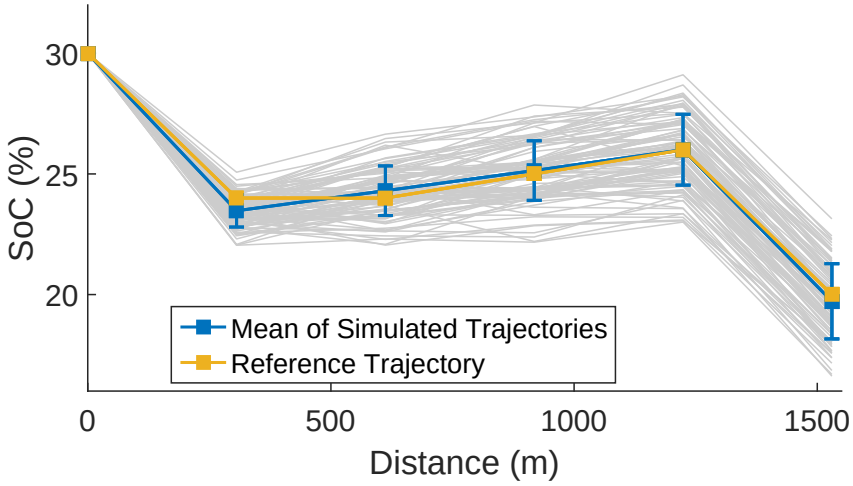
Figure 5.5 – Eco-path (left) and Fastest Path (right) - $(SoC_i = 30\%, SoC_f = 20\%)$

Fig. 5.5 and Table 5.2 show the same results for a final constraint $SoC_f = 20\%$. The consumption is still one third of the one of the fastest path, but now with a more moderate increase in travel distance and time. Also notice that the final SoC constraint

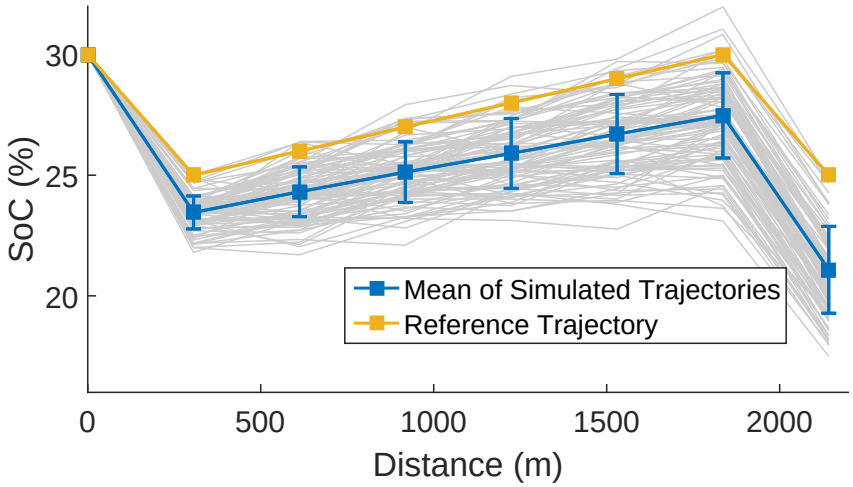
Path	Value	Cons. (L)	SoC_f (%)	Time (s)	Dist. (km)
Eco	0.28	0.013	19.7	106	1.53
Fastest	1.46	0.036	4	44	1.22

Table 5.2 – Comparison of Eco and Fastest Path - $(SoC_i = 30\%, SoC_f = 20\%)$.

is well satisfied in this case by the eco-path, reducing the gap between the resimulated consumption and the value of the eco-routing problem. All in all, we observe a classical trade-off, with the eco-routing having a lower consumption and better management of the state of charge, at the expense of choosing longer and slower paths.



(a) $\Delta SoC = -10\%$



(b) $\Delta SoC = -5\%$

Figure 5.6 – Simulated and reference SoC trajectories. Markers show the mean and vertical bars show $\pm\sigma$ of the simulated trajectories.

Now, we study the SoC a bit more in detail, with Fig.5.6 showing the evolution of the state of charge along the travel. Orange line is the reference SoC from the eco-routing solution, while blue line is the average (with standard deviation indicators) of the 1000 resimulations, with the first 100 resimulations also plotted in grey lines. We see that

for the condition $\Delta SoC = -10\%$ (left), the reference SoC is very well matched by the set of resimulations, which means that the control decisions from the micro problems are able to satisfy the individual final SoC constraints at the end of each segment. In this case the penalty terms are typically close to zero, and the value function is close to the consumption. For the stricter condition $\Delta SoC = -5\%$ (right), we see that the micro solutions begin to have difficulties to reach the required SoC, which leads to penalty terms in the value function (as seen in Table 5.1 and Table 5.2) and a growing gap between the reference SoC from the eco-path and the actual SoC trajectory from the resimulated travels.

We look a bit more into the influence of the final SoC condition, by solving the previous test case for ΔSoC ranging from -10% to $+5\%$. We compare in particular the final SoC and consumption for the eco-path and the resimulated trajectories. Fig.5.7 shows the final SoC (eco-path reference and resimulations average with std indicators). The reference final SoC basically corresponds to the ΔSoC constraint (we recall that $SoC_i = 30\%$). As for the resimulated SoC, we see that for easier constraints such as 10% discharge, it coincides well with the reference, as already seen above in Fig.5.6. When the final SoC condition tightens, we see an increasing gap between the two curves, due to the fact that some of the micro solutions selected for the eco-path do not satisfy their prescribed SoCs. A way to reduce these gaps could be to use a smaller discretization for the initial and final SoC of the micro problems, enabling the routing algorithm to choose reference SoCs closer to the maximum feasible ΔSoC on the segments. Note also that for sufficiently long travels with more opportunities for recharging the battery, the gap may be compensated along the way, see the simulations for the 'macro' optimization in chapter 4.

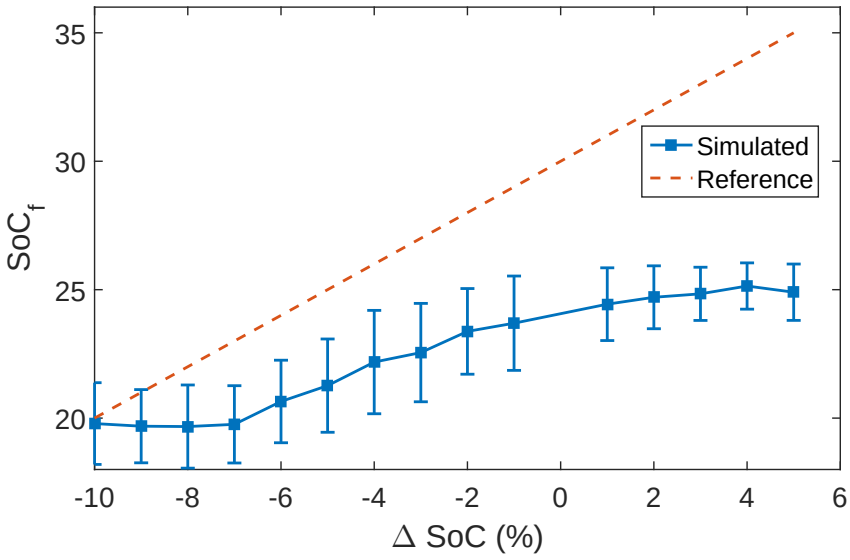


Figure 5.7 – Reference and resimulated SoC_f

On Fig. 5.8 we display the resimulated consumption for the eco and fastest paths. We observe a clear advantage of the eco-paths overall, with a consumption between one quarter and one third of the fastest path. The consumption for the fastest path tends to increase with respect to the ΔSoC, since the harder final SoC constraint required an

additional use of the engine. On the other hand, the consumption of the eco-path appears non increasing, which is probably explained by the fact that the eco-path maintains a low consumption at the expense of an increasing violation of the final SoC constraints, see Fig. 5.7. This is related to the fact that the micro problems manage the reference SoC constraints thanks to the penalization term, which allows some trade-off between the consumption and the reference SoC constraints.

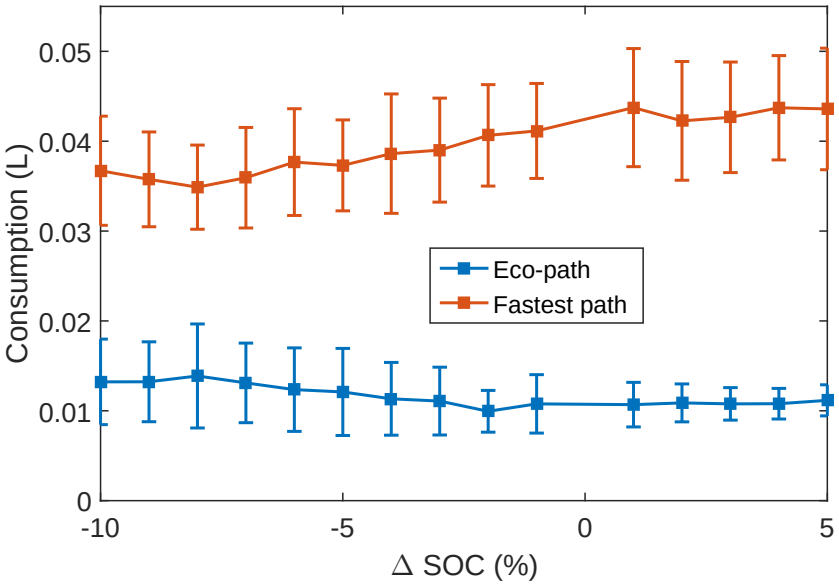


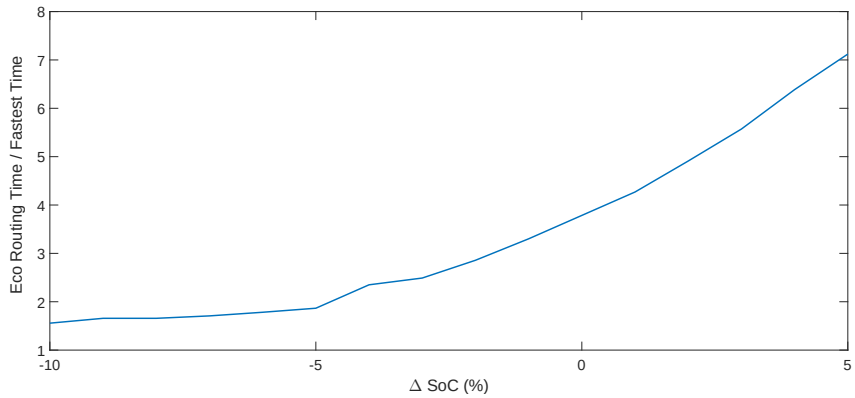
Figure 5.8 – Eco-path and Fastest path Consumption

5.5.2 Study of all possible travels

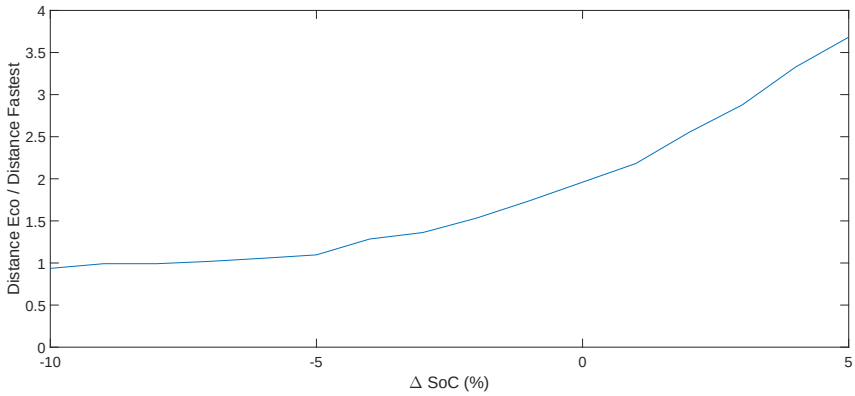
While we previously focused on a single travel, we now perform some simulations while taking into account all possible travels on the road graph. According to the symmetry of the road network, we end up with a set of 54 travels which we solve for varying final SoC conditions. Fig.5.9 shows the average time and distance ratio between the eco-path and the fastest path (i.e. the average of the ratios for each individual travel).

For harder final SoC constraints, we notice a large difference between the two paths, with the eco-path being (on average) up to 7 times slower and 4 times longer than the fastest path. This behaviour comes from the eco-path traveling repeatedly along segments that allow for recharging the battery, in order to meet the final constraint of a 5% charge. Note that this type of path seems to includes cycle in the 'physical' road graph, but not cycles in the weighted graph augmented with the SoC values.

When the final SoC constraints are easier, the average time and distance of the eco-path and fastest path tend to be closer. Indeed, allowing an increased discharge of the battery will reduce the consumption of the fastest path, up to the point that it actually becomes identical to the eco-path.



(a) Time Ratio



(b) Distance Ratio

Figure 5.9 – Time and Distance Ratio

5.6 Conclusion

Simulations on a simple road network indicate that the eco-routing method computes optimal paths with a consumption significantly lower than the fastest path solutions. This reduced consumption comes with an expected trade-off in terms of travel distance and time, and considering an upper bound for time and/or distance would be a natural extension of the method, in order to obtain solutions that can fit the expectations of the drivers. We also observe that the accuracy of the reference SoC trajectory computed by the eco-routing tends to decrease for stricter final SoC conditions. Increasing the weight of the penalty term for the final SoC constraint would also improve this accuracy, however at the expense of a higher consumption overall. Another direction for improvement would be to use a finer SoC discretization for the initial and final conditions of the micro problems, as the error analysis has shown that the values of the discretized problem approximate the one of the original problem up to the order of the discretization step size.

CHAPTER 6

Conclusions and Perspectives

Contents

In this chapter, the main results obtained through this thesis are recalled. Extensions and perspectives are finally presented in the last section.

6.1 Conclusions

As a first step, a stochastic data-based model of traffic has been presented. Taking into account a division of the road networks into segments, this model reduces the speed/acceleration data recorded from the vehicles on each segment. This approach has shown its ability to estimate the distribution of the energy used by the vehicles on the road segment. This statistical approach, using probability distributions, is able to significantly reduce large amounts of traffic data. Using clustering techniques, based on optimal transport theory, we can reach compression factor up to 99% while retaining most relevant information. These traffic distributions can be computed off-line and stored in databases. Numerical experiments, with simulated traffic data, have shown that the method can be used on large networks in reasonable time, the city of Luxembourg in few hours. Besides, experiments on real traffic data have shown that the obtained clusters are efficient for identifying different traffic conditions. The study on the influence of discretization seems to indicate there is an expected trade-off between the support size of the speed/acceleration distribution and the accuracy of the estimated energy distributions.

Using this stochastic model of traffic, we have been able to optimize, given initial and final SoC, the energy consumption of HEVs on a road segment under traffic conditions. A stochastic optimization problem has been introduced and solved using Stochastic Dynamic Programming algorithm. The solution, compared to a theoretical lower bound obtained for specific speed profiles by Deterministic Dynamic Programming algorithm, shows that the consumption is at less than 5% from the lower bound. The optimal policies can be interpreted in terms of operating modes and show that the vehicle adapts the torque of the engine according both to the desired final state of charge and traffic conditions. Using previously computed and stored traffic distributions, it is possible to solve these problems off-line for all road segments and store the resulting cost maps and optimal policies. This information can be used on-board of the vehicle, for example through a rule-based energy management system.

The stochastic problem is usually too costly to be solved on-line on the whole travel. Therefore we introduced a bi-level formulation. Such a formulation enables us to handle the stochastic aspects of the traffic at the lower level, using the previously mentioned stochastic optimization approach. On the other hand, the upper level can then be formulated as a deterministic optimization problem and solved thanks to Deterministic Dynamic Programming algorithm. This bi-level method has shown its capacity to determine a reference state of charge trajectory on a whole travel, with computational time reasonable for on-line purposes. This computational time of the upper level does not depend on the size of the support of the traffic distributions, allowing to use finer discretizations to improve accuracy of the energy cost maps. Numerical simulations indicate an average overconsumption, around 15%, compared to the theoretical lower bound with perfect speed profile prediction.

Finer speed and acceleration discretizations increase the support size for the traffic distributions, leading to a more expensive clustering step (performed off-line) . As an alternative to the usual Sinkhorn iterations, we adapted a stochastic gradient algorithm to the Wasserstein barycenter problem. Thanks to the entropic regularizations and the

semi-dual formulation, we obtained an approximate barycenter problem with an objective function that has an additive structure. This algorithm has an increased space occupancy, but a reduced time complexity with a linear convergence rate. Using such an algorithm can reduce the computational time of clustering on fine speed/acceleration discrete grids.

6.2 Perspectives

We have proposed a memoryless method that does not keep track of the temporal dependency of the speed profile. Even if this assumption gives good results in terms of consumption estimation, it generates unrealistic speed profiles, typically discontinuous. It may be possible to improve the estimated consumption with a Markovian approach that keeps the time dependency between the speed and acceleration. The main difficulty of such a modelisation will be in the clustering part that needs to preserve the dynamic aspects of the aggregated data.

Another possible improvement of the method is the determination of the optimal number of clusters needed to represent the traffic conditions on the road segment. Approaches such as the silhouette method may help to determine the optimal number of clusters.

A current limitation is that recorded data are not linked to the speed and acceleration characteristics of vehicles. It may be interesting to aggregate the data according to the type of vehicle (light vehicles, trucks, buses, etc.), in order to improve the consumption estimation.

Recovering the vehicles data may be difficult to implement over the entire road network, and lack of information may appear on some parts of it. Meta-clustering, i.e. performing a second layer of clustering on the road segments with respect to their clustered traffic conditions, may solve the sparsity of the data. Indeed, such clustering may highlight the link between the topological characteristic of the road segments (slope, curve, type of intersection, etc.) and typical traffic conditions.

The assumption that the vehicle follows the traffic, i.e. the speed and the acceleration are i.i.d with respect to the traffic distribution, does not allow the driver to choose his speed. In the case of autonomous vehicles, relaxing the traffic assumption and adding the speed to the decision variables may be considered in future works. Such an approach may merge EMS, eco-driving, and eco-routing.

In our work, time stationary traffic conditions have been considered, which seems rather unrealistic for long travels. Adding the time as an additional state variable is likely to be too costly due to the curse of dimensionality. Besides, in the case of stochastic transitions on traffic distributions then the deterministic ‘macro’ level can be transformed into a stochastic problem.

More generally, additional constraints on the vehicle can be considered. For example, recent works on the ageing of batteries have shown that from an economical point of view, the fuel saving generated thanks to the hybridization, can be outweighed by the premature ageing of the battery [29]. Adding an ageing constraint or cost in the ’micro’ problem could help to solve this issue.

APPENDIX A

A Stochastic gradient method for Wasserstein barycenters of distributions

Contents

A.1 Introduction

Let μ and ν be probability measures over discrete sets I and J, and c_{ij}, for $i \in I$ and $j \in J$, represent the transportation cost from i to j. An optimal transport plan π from mu over ν is defined as a solution of the 'Monge-Kantorovich' linear programming problem [66],

$$W(\mu, \nu) = \min_{\pi \in \Pi(\mu,\nu)} d \cdot \pi \tag{A.1.0.1}$$

Here the set of transportation plans $\Pi(\mu, \nu)$ is defined as the set of nonnegative matrices of size $|I| \times |J$ with marginals $\pi^\top \mathbf{1} = \mu$ and $\pi \mathbf{1} = \nu$. Since the feasible set is nonempty and bounded, by the linear programming theory, the set of solutions is nonempty and the optimal value is the same as the one of the dual problem.

Solving this problem by linear programming algorithms may be expensive for large scale problems (the basis matrix has size $|I| + |J|$). So the standard practice is based on entropic regularization and Sinkhorn iterations based on iterative Bregmann projections, see [8]. The regularized problem may also be solved by the Stochastic Average Gradient approach [4].

In some cases of transportation costs based on the Euclidean distance of points x^i and y^j, elements of the same Euclidean space, we obtain the so-called Wasserstein distance. Minimizing the convex combination of distances to given distributions, an extension of the notion of barycenters, called Wasserstein barycenter, has been obtained [1] in the case of $p = 2$.

From a practical point of view, the computation of Wasserstein barycenters is computationally expensive. To decrease the complexity of the computation of the Wasserstein distance, some methods based on 1-D projection have been proposed, such that the sliced method [78] or the Radon transform [9]. Another direction is to discretize the state space and use linear programming techniques, dualization and descent gradient algorithm [19] or quasi-Newton methods [13]. In this article, we propose a stochastic gradient algorithm for the computation of Wasserstein barycenters in the discrete case. This method retains the same rate of convergence as the Sinkhorn iterations with a smaller complexity.

A.2 Barycenter type Optimal Transportation

A.2.1 Framework

Let K be a positive integer, and X^0 to X^K be discrete sets, with X^k of size n_k. For $k = 1$ to K, X^k is endowed with a probability ν^k, and the transportation cost from $i \in X^0$ to $j \in X^k$ is denoted by c^k_{ij}; so, c^k is a $n_0 \times I^k$ matrix. We say that $\pi = (\pi^k)_{1\ldots K}$, with π^k matrix of same size as c^k, is a transport plan if it is a probability law over $X^0 \times X^k$, with marginals ν^k, for $k = 1$ to K, and the same marginal over X^0 for $k = 1$ to K. We may assume that all components of the ν_k are positive (otherwise we could reformulate the problem by eliminating zero components). The associated cost is

$$c \cdot \pi = \sum_{k=1}^{K} c^k \cdot \pi^k \tag{A.2.1.1}$$

The minimal cost transportation plan (where the marginal over X^0, denoted by μ, is to be minimized) is therefore solution of the problem

$$\begin{array}{l} \min_{\pi,\mu} c \cdot \pi; \;\; \pi \geq 0; \\ (\pi^k)^\top \mathbf{1} = \mu, \;\; \pi^k \mathbf{1} = \nu^k, \;\; k = 1, \ldots, K, \end{array} \tag{$\mathcal{P}$}$$

Here $\mathbf{1}$ denotes the vector of ones (with dimension coherent with the matrix/vector products).

A.2.2 Penalized formulations

For the moment a regularization function is just a l.s.c. convex proper functions over $\mathcal{R}$. Such a regularization function e is equal to its bi-conjugate; that is, it can be represented as

$$e(s) = \sup_t st - e^*(t), \tag{A.2.2.1}$$

where $e^* : \mathcal{R} \to \bar{\mathcal{R}}$ is the *Fenchel conjugate* defined by

$$e^*(t) := \sup_s st - e(s). \tag{A.2.2.2}$$

We keep as much as possible the framework with general penalties, but later we will specialize in the case of the reverse entropy, with domain $\mathcal{R}_+$. whose expression and Fenchel conjugate are

$$\varphi(s) = s(\log s - 1); \quad \varphi^*(t) := e^t. \tag{A.2.2.3}$$

We choose to regularize both π and μ with some regularization functions e_1 and e_2: and set

$$E_1(\pi) := \sum_{k=1}^{K} \sum_{i \in X^0} \sum_{j \in X^k} e_1(\pi_{ij}^k); \quad E_2(\mu) := \sum_{i \in X^0} e_2(\mu_i). \tag{A.2.2.4}$$

We choose e_1 with domain either $\mathcal{R}_+$ or $(0, \infty)$, differentiable over $(0, \infty)$; therefore it can be interpreted as a way to ensure the non negativity of π.

Then the *regularized transportation problem* is, given regularization parameters $\varepsilon_1 \geq 0$ and $\varepsilon_2 \geq 0$:

$$\begin{array}{c} \min_{\pi,\mu} c \cdot \pi + \varepsilon_1 E(\pi) + \varepsilon_2 E(\mu); \\ \pi^k \mathbf{1} = \mu; \quad (\pi^k)^\top \mathbf{1} = \nu^k, \;\; k = 1, \ldots, K. \end{array} \tag{$\mathcal{P}_\varepsilon$}$$

In agreement with the convention in convex analysis, we interpret a zero penalization term as the limiting one when the penalty parameter goes to zero, that is, for e_1, as the indicatrix of $\mathcal{R}_+$ (so that the non negativity constraint is always valid). The *Lagrangian* of $\mathcal{P}_\varepsilon$ is

$$\begin{array}{rl} \mathcal{L}_\varepsilon(\pi, \mu, u, v) := & c \cdot \pi + \varepsilon_1 E(\pi) + \varepsilon_2 E(\mu) \\ & + \sum_{k=1}^{K} \left(u^k \cdot ((\pi^k)^\top \mathbf{1} - \mu) + v^k \cdot (\pi^k \mathbf{1} - \nu^k) \right) \end{array} \tag{$\mathcal{L}_\varepsilon$}$$

Here $(u^k)_{\{1,\dots,K\}} \in \mathbb{R}^I$ and $(v^k)_{\{1,\dots,K\}} \in \mathbb{R}^J$ are the *Lagrange multipliers* associated with the linear constraints of $\mathcal{P}_\varepsilon$. Equivalently

$$\begin{aligned}\mathcal{L}_\varepsilon(\pi,\mu,u,v) := & \sum_{k=1}^{K}\sum_{i\in X^0}\sum_{j\in X^k}\left(\varepsilon_1 e_1(\pi_{ij}^k)+\pi_{ij}^k(c_{ij}^k+u_i^k+v_j^k)\right)\\ & +\sum_{i\in X^0}\left(\varepsilon_2 e_2(\mu_i)-\mu_i\sum_{k=1}^{K}u_i^k\right)-v\cdot\nu.\end{aligned} \tag{A.2.2.5}$$

Keep in mind the definition (A.2.2.1) of the Fenchel conjugate, implying that for $\kappa > 0$:

$$(\kappa e)^*(s) = \kappa(\sup_t((\kappa^{-1}s)t - e(t)) = \kappa e^*(\kappa^{-1}s). \tag{A.2.2.6}$$

We see that the *dual problem*, i.e. the one of maximizing w.r.t. the Lagrange multipliers the infimum of $\mathcal{L}_\varepsilon(\pi,\mu,u,v)$ w.r.t. the primal variables (π,μ), is

$$\max_{u,v} -v\cdot\nu-\varepsilon_1\sum_{k=1}^{K}\sum_{i\in X^0}\sum_{j\in X^k}e_1^*\left(-\varepsilon_1^{-1}(c_{ij}^k+u_i^k+v_j^k)\right)-\varepsilon_2\sum_{i\in X^0}e_2^*\left(\frac{1}{\varepsilon_2}\sum_{k=1}^{K}u_i^k\right) \quad (\mathcal{D}_\varepsilon)$$

Remark 7. Since the non negativity of π implies the one of μ, we may as well decide not to penalize the nonnegativity of the latter, which amounts to take for e_2 the null function. Then e_2^* is the indicatrix of 0, so that the expression of the dual problem is

$$\max_{u,v} -v\cdot\nu-\varepsilon_1\sum_{k=1}^{K}\sum_{i\in X^0}\sum_{j\in X^k}e_1^*\left(-\varepsilon_1^{-1}(c_{ij}^k+u_i^k+v_j^k)\right);\quad -\sum_{k=1}^{K}u^k=0. \tag{A.2.2.7}$$

A.3 Semi-dual problem

Let us denote the above dual cost by $D(u,v)$, defined in A.2.2.7. We define the *semi-dual cost* as

$$d(u) := \sup_v D(u,v) \tag{A.3.0.1}$$

and by *semi-dual problem* the problem

$$\max_u d(u). \tag{A.3.0.2}$$

Clearly the dual and semi-dual problem have the same value. We next assume that

$$e_1 = \varphi \tag{A.3.0.3}$$

where the reverse entropy φ was defined in (A.2.2.3). Then e_1^* is the exponential function. This allows to get an explicit expression of the semi-dual cost. Indeed we maximize a concave function of v. Setting $\xi_{ij}^k := e^{-c_{ij}^k/\varepsilon_1}$, we see that the maximum is attained w.r.t. v_j^k iff

$$\sum_{i\in X^0}\xi_{ij}^k e^{\frac{-u_i^k-v_j^k}{\varepsilon_1}} = \nu_j^k. \tag{A.3.0.4}$$

Remember that $\nu_j^k > 0$, so that the above equation has a unique solution

$$v_j^k = \varepsilon_1 \log(\frac{1}{\nu_j^k} \sum_{i \in X^0} \xi_{ij}^k e^{\frac{-u_i^k}{\varepsilon_1}}). \tag{A.3.0.5}$$

We obtain the expression of the semi-dual problem:

$$\max_u -\varepsilon_1 \sum_{k=1}^{K} \sum_{j \in X^k} \nu_j^k (\log(\frac{1}{\nu_j^k} \sum_{i \in X^0} \xi_{ij}^k e^{\frac{-u_i^k}{\varepsilon_1}}) + 1) - \varepsilon_2 \sum_{i \in X^0} e_2^* \left(\frac{1}{\varepsilon_2} \sum_{k=1}^{K} u_i^k \right). \tag{A.3.0.6}$$

In particular when $e_2 = \varphi$ we obtain

$$\max_u -\varepsilon_1 \sum_{k=1}^{K} \sum_{j \in X^k} \nu_j^k (\log(\frac{1}{\nu_j^k} \sum_{i \in X^0} \xi_{ij}^k e^{\frac{-u_i^k}{\varepsilon_1}}) + 1) - \varepsilon_2 \sum_{i \in X^0} e^{\frac{\sum_{r=1}^{K} u_i^r}{\varepsilon_2}} \tag{A.3.0.7}$$

Since the ν^k are probability distributions we may rewrite the problem in the form

$$\max_u \sum_{k=1}^{K} \sum_{j \in X^k} h_k^j(u) \nu_j^k \tag{$\mathcal{S}_{\varepsilon_1,\varepsilon_2}$}$$

where

$$h_k^j(u) = -\varepsilon_1 \log(\frac{1}{\nu_j^k} \sum_{i \in X^0} \xi_{ij}^k e^{-u_i^k/\varepsilon_1}) - \varepsilon_1 - \frac{\varepsilon_2}{K} \sum_{i \in X^0} e^{\frac{\sum_{r=1}^{K} u_i^r}{\varepsilon_2}} \tag{A.3.0.8}$$

The expression of the gradient of h_k^j is

$$\forall p \in [1, K],\ \nabla_{u_i^p} h_k^j(u) = -\frac{e^{\frac{\sum_{r=1}^{K} u_i^r}{\varepsilon_2}}}{K} + \delta_{k,p} \frac{\xi_{ij}^k e^{-u_i^k/\varepsilon_1}}{\sum_{l \in X^0} \xi_{lj}^k e^{-u_l^k/\varepsilon_1}} \tag{A.3.0.9}$$

Other way keeping the sum over K inside the functions, we obtain :

$$\max_u \sum_j h^j(u) \tag{$\mathcal{S}_{\varepsilon_1,\varepsilon_2}$}$$

with :

$$h^j(u) = -\varepsilon_1 \sum_k \nu_j^k (\log(\frac{1}{\nu_j^k} \sum_{i \in X^0} \xi_{ij}^k e^{\frac{-u_i^k}{\varepsilon_1}}) + 1) - \frac{\varepsilon_2 \sum_k \nu_j^k}{K} (\sum_{i \in X^0} e^{\frac{\sum_{r=1}^{K} u_i^r}{\varepsilon_2}}) \tag{A.3.0.10}$$

We can compute the gradient of h :

$$\forall k \in [1, K], \nabla_{u_i^k} h^j(u) = -\frac{1}{K} \sum_{k=1}^{K} \nu_j^k e^{\frac{\sum_{r=1}^{K} u_i^r}{\varepsilon_2}} + \frac{\nu_j^k \xi_{ij}^k e^{-\frac{u_i^k}{\varepsilon_1}}}{\sum_l \xi_{lj}^k e^{-\frac{u_l^k}{\varepsilon_1}}} \tag{A.3.0.11}$$

A.4 Algorithms

In the following section, we consider algorithms computing, for K measures of probability ν^k with support of size I, their barycenter μ having a support of size J. For the sake of simplification, the data of the problem $(\nu^k, \mu, \xi^k, \varepsilon_1, \varepsilon_2)$ are not listed in the inputs of the algorithms.

A.4.1 SAG

Algorithm 8: SAG for Discrete OT

Input : $\rho \in \mathbb{R}^{+*}$,$\eta \in \mathbb{R}^{+*}$
Output: $\mathbf{u} \in \mathbb{R}^{K \times I}$
Initialization
$\mathbf{u} \leftarrow 0_{K \times I}$, $\mathbf{d} \leftarrow 0_{K \times I}$, $\forall j \in \{1, .., J\}, \mathbf{g}_j \leftarrow 0_{K \times I}$;
while $\|\nabla h\| \leq \eta$ **do**
  Draw $j \in \{1, 2, ..., J\}$ uniformly.
  $\mathbf{d} \leftarrow \mathbf{d} - \mathbf{g}_j$
  $\mathbf{g}_j \leftarrow \nabla h(x_j, \mathbf{u})$ **% see Algorithm 2**
  $\mathbf{d} \leftarrow \mathbf{d} + \mathbf{g}_j$
  $\mathbf{u} \leftarrow \mathbf{u} + \rho \mathbf{d}$

Algorithm 9: Gradient Computation

Input : $j \in \{1, \ldots, J\}$, $u \in \mathbb{R}^{K \times I}$
Output: $\nabla h(x_j, u)$
Initialization
$\bar{\nu} \leftarrow \sum_{r \in \{1,\ldots,K\}} \nu_j^r$
$\bar{u} \leftarrow \sum_{r \in \{1,\ldots,K\}} u^r$
for $k = 1, \ldots, K$ **do**
  $D \leftarrow \sum_{l \in \{1,\ldots,I\}} \xi_{lj}^k e^{-\frac{u_l^k}{\varepsilon_1}}$
  for $i = 1, \ldots, I$ **do**
    $\nabla_{u_i^k} h(x_j, u) \leftarrow -\frac{\bar{\nu} e^{\frac{\bar{u}_i}{\varepsilon_2}}}{K} + \frac{\nu_j^k \xi_{ij}^k e^{-\frac{u_i^k}{\varepsilon_1}}}{D}$

The complexity of the SAG for each iteration is $O(KI)$. Indeed at each step, the algorithm computes the gradient of a specific h_j. We can see from algorithm 9 that each component of the gradient needs only $2K + 2KI$ operations as the two sum, $\bar{u}, D$ can be compute in a first time. So we need at most $O(KI)$ operations to complete a step. In terms of memory, we store J matrices of size KI for the gradient, so the space complexity is $O(KIJ)$

A.4.2 Sinkhorn

We introduce the notation $\xi = e^{-\frac{c}{\varepsilon_1}} \in \mathbb{R}^{I \times J}$ by analogy with $\xi^k = e^{-\frac{c^k}{\varepsilon_1}}$. And the two operators $\odot, \div$ component-wise multiplication and division.

Algorithm 10: Sinkhorn for Discrete OT

Output: μ
Initialization
for $k \in 1, ..., K$ **do**
  $x_k \leftarrow \mathbf{1}_J$
  $y_k \leftarrow \nu^k \div (\xi^\top \mathbf{1}_I)$
for $n = 1, 2, ..., N$ **do**
  $\mu \leftarrow \prod_{k \in \{1,...,K\}} (x_k \odot (\xi y_k))^{\lambda_k}$
  for $k \in 1, ..., K$ **do**
    $x_k \leftarrow \mu \div (\xi \mathrm{diag}(y_k)\mathbf{1}_I)$
    $y_k \leftarrow \nu^k \div (\xi^\top \mathrm{diag}(x_k)\mathbf{1}_J)$
$\mu \leftarrow \prod_{k \in \{1,...,K\}} (x_k \odot (\xi y_k))^{\lambda_k}$

We can see that at each step we have a computation of μ and a loop over K. The computation of μ is $O(KIJ)$. The key loop updates x_k and y_k for a complexity of $O(KI^2J + KIJ^2)$ [1]. So we have a complexity of the Sinkhorn algorithm bounded by $O(N^4)$ where $N = \max(K, I, J)$. In terms of space, we store K vectors x_k of size J, K vectors y_k of size I, and a matrix of size IJ. So we have a space complexity in $O(KI + KJ + IJ)$.

Algorithm	SAG	Sinkhorn
Time complexity per iteration	$O(KI) \leq O(N^2)$	$O(KI^2J + KIJ^2) \leq O(N^4)$
Space complexity	$O(KIJ) \leq O(N^3)$	$O(KI + KJ + IJ) \leq O(N^2)$

Table A.1 – Complexity of Algorithms

A.5 Convergence Rate

A.5.1 SAG algorithm

In [79], the following results over the convergence rate of SAG have been proved. Let assumed (i) that each functions $h(x_j, \cdot)$ is differentiable and each gradient $\nabla h(x_j, \cdot)$ is Lipschitz continuous with constant L. And (ii) that the function $F(\cdot) = \sum_{j \in \{1,...,J\}} h(x_j, \cdot)$ is strongly convex with constant β [2]. A consequence of (ii) is the existence of a unique optimum u^*. The following results holds.

Proposition 8. *With constant step size of $\rho = \frac{1}{2JL}$, the SAG iterations satisfy for $n > 1$:*

$$\mathbb{E}[\|u^k - u^*\|^2] \leq (1 - \frac{\beta}{8LJ})^n [3\|u^0 - u^*\|^2 + 9\frac{\sigma^2}{4L^2}]$$

[1] Assuming that the complexity of a product of a matrix $(m \times n)$ by a matrix of size $(n \times p)$ is $O(nmp)$
[2] Meaning that $x \mapsto g(x) - \frac{\beta}{2}\|x\|^2$ is convex

with $\sigma^2 = \sum_{j\in\{1,\dots,J\}} \|\nabla h(x_j, u^*)\|^2$.

This gives a convergence rate of $O(J/K)$ and the linear convergence holds for any $\rho \leq \frac{1}{2JL}$.

A.5.2 The Sinkhorn Algorithm

The Sinkhorn algorithm [82] is originally a matrix scaling method based on alternative scaling of rows and columns in order to obtain a doubly stochastic matrix, or more generally, a nonnegative matrix with given marginals. Convergence rates are known [37]. The rate can be estimated as follows. If a is the positive matrix to be scaled, set

$$\theta(a) := \max_{i,j,k,\ell} \log \frac{a_{ik}a_{j\ell}}{a_{i\ell}a_{jk}}; \quad \kappa(a) : \frac{e^{\frac{1}{2}\theta(a)} - 1}{e^{\frac{1}{2}\theta(a)} + 1} \tag{A.5.2.1}$$

then the rate is $\gamma := \kappa(a)^2$, for some special metric.

It appears that iterative Bregman projections for solving a optimal transport problem coincide with the Sinkhorn algorithm for scaling a related matrix, see [18].

It was shown in [8] that Bregman projections naturally apply to the barycenter problem, the essential steps being based in scaling steps similar to those in the Sinkhorn algorithm. Since the Bregman projections are over affine sets, the sequence of iterates converges, see[11], but the convergence rate is unknown.

Bibliography

[1] M. Agueh and G. Carlier, "Barycenters in the Wasserstein space," *SIAM Journal on Mathematical Analysis*, vol. 43, no. 2, pp. 904–924, SIAM, 2011.

[2] K. Ahn and H. Rakha, "Field Evaluation of Energy and Environmental Impacts of Driver Route Choice Decisions," *2007 IEEE Intelligent Transportation Systems Conference*, pp. 730–735, 2007.

[3] *Ambient air pollution: a global assessment of exposure and burden of disease*, World Health Organization, 2016.

[4] G. Aude, M. Cuturi, G. Peyré, and F. Bach, "Stochastic Optimization for Large-scale Optimal Transport," *arXiv preprint arXiv:1605.08527*, 2016.

[5] M. Bardi and I. Capuzzo-Dolcetta, *Optimal control and viscosity solutions of Hamilton-Jacobi-Bellman equations*, Birkhäuser, Boston, MA, 1997, pp. xviii+570.

[6] M. Barth, K. Boriboonsomsin, and A. Vu, "Environmentally-Friendly Navigation," *2007 IEEE Intelligent Transportation Systems Conference*, pp. 684–689, 2007.

[7] R. Bellman, "On a routing problem," *Quarterly of applied mathematics*, vol. 16, no. 1, pp. 87–90, 1958.

[8] J.-D. Benamou, G. Carlier, M. Cuturi, L. Nenna, and G. Peyré, "Iterative Bregman projections for regularized transportation problems," *SIAM J. Sci. Comput.* Vol. 37, no. 2, A1111–A1138, 2015.

[9] N. Bonneel, J. Rabin, G. Peyré, and H. Pfister, "Sliced and Radon Wasserstein barycenters of measures," *J. Math. Imaging Vision*, vol. 51, no. 1, pp. 22–45, 2015.

[10] K. R. Bouwman, T. H. Pham, S Wilkins, and T Hofman, "Predictive Energy Management Strategy Including Traffic Flow Data for Hybrid Electric Vehicles," *IFAC PapersOnLine*, vol. 50, no. 1, pp. 10046–10051, Elsevier B.V., 2017.

[11] L. M. Bregman, "The relaxation method of finding the common point of convex sets and its application to the solution of problems in convex programming," *USSR computational mathematics and mathematical physics*, vol. 7, no. 3, pp. 200–217, Elsevier, 1967.

[12] P. Cao, T. Miwa, and T. Morikawa, "Use of Probe Vehicle Data to Determine Joint Probability Distributions of Vehicle Location and Speed on an Arterial Road," *Transportation Research Record*, vol. 2421, no. 1, 2014.

[13] G. Carlier, A. Oberman, and E. Oudet, "Numerical methods for matching for teams and Wasserstein barycenters," *ESAIM Math. Model. Numer. Anal.* Vol. 49, no. 6, pp. 1621–1642, 2015.

[14] Chan-Chiao Lin, Huei Peng, and J. W. Grizzle, "A stochastic control strategy for hybrid electric vehicles," *Proceedings of the 2004 American Control Conference*, 4710–4715 vol.5, 2004.

[15] K. T. Chau and Y. S. Wong, "Overview of power management in hybrid electric vehicles," *Energy Conversion and Management*, vol. 43, no. 15, pp. 1953–1968, 2002.

[16] L. Codecá, R. Frank, S. Faye, and T. Engel, "Luxembourg SUMO Traffic (LuST) Scenario: Traffic Demand Evaluation," *IEEE Intelligent Transportation Systems Magazine*, vol. 9, no. 2, pp. 52–63, IEEE, 2017.

[17] T. H. Cormen, *Introduction to algorithms*, MIT press, 2009.

[18] M. Cuturi, "Sinkhorn Distances: Lightspeed Computation of Optimal Transport," 2013.

[19] M. Cuturi and A. Doucet, "Fast Computation of Wasserstein Barycenters," *Proceedings of the 31st International Conference on Machine Learning*, pp. 685–693, Bejing, China: PMLR, 2014.

[20] I. Dagan, L. Lee, and F. Pereira, "Similarity-based Methods for Word Sense Disambiguation," *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics and Eighth Conference of the European Chapter of the Association for Computational Linguistics*, pp. 56–63, Madrid, Spain: Association for Computational Linguistics, 1997.

[21] G. De Nunzio, G. Gomes, C. Canudas-de-Wit, R. Horowitz, and P. Moulin, "Speed Advisory and Signal Offsets Control for Arterial Bandwidth Maximization and Energy Consumption Reduction," *IEEE Transactions on Control Systems Technology*, vol. 25, no. 3, pp. 875–887, 2017.

[22] G. De Nunzio and L. Thibault, "Energy-optimal driving range prediction for electric vehicles," *Intelligent Vehicles Symposium (IV), 2017 IEEE*, pp. 1608–1613, 2017.

[23] G. De Nunzio, L. Thibault, and A. Sciarretta, "Model-based eco-routing strategy for electric vehicles in large urban networks," *Comprehensive Energy Management–Eco Routing & Velocity Profiles*, pp. 81–99, Springer, 2017.

[24] I. De Vlieger, D. De Keukeleere, and J. G. Kretzschmar, "Environmental effects of driving behaviour and congestion related to passenger cars," *Atmospheric Environment*, vol. 34, pp. 4649–4655, 2000.

[25] E. W. Dijkstra, "A note on two problems in connexion with graphs," *Numerische mathematik*, vol. 1, no. 1, pp. 269–271, Springer, 1959.

[26] R. L. Dobrushin, "Prescribing a System of Random Variables by Conditional Distributions," *Theory of Probability & Its Applications*, vol. 15, no. 3, pp. 458–486, 1970.

[27] *Driving into 2025: The Future of Electric Vehicles*, J.P. Morgan.

[28] Y. Dujardin, D. Vanderpooten, and F. Boillot, "A multi-objective interactive system for adaptive traffic control," *European Journal of Operational Research*, vol. 244, no. 2, pp. 601 –610, 2015.

[29] S. Ebbesen and L. Guzzella, "Trade-Off Between Fuel Economy and Battery Life for Hybrid Electric Vehicles," *Dynamic Systems and Control Conference*, vol. 2, no. 2, pp. 217–223, Oct. 2011.

[30] D. M. Endres and J. E. Schindelin, "A new metric for probability distributions," *IEEE Transactions on Information Theory*, vol. 49, no. 7, pp. 1858–1860, 2003.

[31] E. Ericsson, "Variability in urban driving patterns," *Transportation Research Part D: Transport and Environment*, vol. 5, no. 5, pp. 337 –354, 2000.

[32] E. Ericsson, H. Larsson, and K. Brundell-Freij, "Optimizing route choice for lowest fuel consumption–potential effects of a new driver support tool," *Transportation Research Part C: Emerging Technologies*, vol. 14, no. 6, pp. 369–383, Elsevier, 2006.

[33] *European Strategies - White Paper 2011*, URL: `https://goo.gl/GoSWcf`.

[34] L. Evans, R. Herman, and T. Lam, "Multivariate Analysis of Traffic Factors Related to Fuel Consumption in Urban Driving," *Transportation Science*, vol. 10, no. 2, pp. 205–215, 1976.

[35] M. Falcone and R. Ferretti, *Semi-Lagrangian Approximation Schemes for Linear and Hamilton-Jacobi Equations*, SIAM Publications, 2013.

[36] C. Fiori, V. Arcidiacono, G. Fontaras, M. Makridis, K. Mattas, V. Marzano, C. Thiel, and B. Ciuffo, "The effect of electrified mobility on the relationship between traffic conditions and energy consumption," *Transportation Research Part D*, vol. 67, pp. 275–290, 2019.

[37] J. Franklin and J. Lorenz, "On the scaling of multidimensional matrices," *Linear Algebra and its Applications*, vol. 114-115, pp. 717–735, North-Holland, 1989.

[38] J. Fu, S. Song, Z. Fu, and J. Ma, "Hierarchical Model Predictive Control for Parallel Hybrid Electric Vehicles," *Asian Journal of Control*, vol. 20, no. 6, pp. 2331–2342, 2018.

[39] X. Fuguo, J. Yuan, and J. Xiaohong, "A Modified Energy Management Strategy Based on SDP Policy Iteration for Commuter Hybrid Electric Vehicles," *2016 35th Chinese Control Conference (CCC)*, pp. 2537–2541, TCCT, 2016.

[40] *Geco air*, URL: `http://www.gecoair.fr/home_en/`.

[41] T. Gnann, T. S. Stephens, Z. Lin, P. Plötz, C. Liu, and J. Brokate, "What drives the market for plug-in electric vehicles ? - A review of international PEV market diffusion models," *Renewable and Sustainable Energy Reviews*, vol. 93, pp. 158–164, Elsevier Ltd, 2018.

[42] C. Guo, B. Yang, O. Andersen, C. S. Jensen, and K. Torp, "EcoMark 2.0: Empowering Eco-Routing with Vehicular Environmental Models and Actual Vehicle Fuel Consumption Data," *Geoinformatica*, vol. 19, pp. 567–599, 2015.

[43] P. E. Hart, N. J. Nilsson, and B. Raphael, "A Formal Basis for the Heuristic Determination of Minimum Cost Paths," *IEEE Transactions on Systems Science and Cybernetics*, vol. 4, no. 2, pp. 100–107, 1968.

[44] J. G. Hayes and K. Davis, "Simplified electric vehicle powertrain model for range and energy consumption based on EPA coast-down parameters and test validation by Argonne National Lab data on the Nissan Leaf," *2014 IEEE Transportation Electrification Conference and Expo (ITEC)*, pp. 1–6, 2014.

[45] Y. Huang, H. Wang, A. Khajepour, H. He, and J. Ji, "Model predictive control power management strategies for HEVs: A review," *Journal of Power Sources*, vol. 341, pp. 91 –106, 2017.

[46] *International Energy Agency*, URL: https://www.iea.org/tcep/transport/.

[47] X. Jiao and T. Shen, "SDP Policy Iteration-Based Energy Management Strategy Using Traffic Information for Commuter Hybrid Electric Vehicles," *Energies*, vol. 7, no. 7, pp. 4648–4675, 2014.

[48] L. Jin, A. A. Kurzhanskiy, and S. Amin, "Throughput-Improving Control of Highways Facing Stochastic Perturbations," *CoRR*, vol. abs/1809.07610, 2018.

[49] D. Karbowski, V. Sokolov, and A. Rousseau, "Vehicle Energy Management Optimization through Digital Maps and Connectivity," *ITS World Congress*, 2015.

[50] M. Kato, "Historical Overview of Electronics and Automobiles: Breakthroughs and Innovation by Electronics and Electrical Technology," *Encyclopedia of Automotive Engineering*, pp. 1–13, Wiley Online Library, 2014.

[51] S. Kermani, S. Delprat, T. M. Guerra, and R. Trigui, "Predictive control for HEV energy management: experimental results," *2009 IEEE Vehicle Power and Propulsion Conference*, pp. 364–369, 2009.

[52] S. Kermani, S. Delprat, R. Trigui, and T. Guerra, "Predictive energy management of hybrid vehicle," *2008 IEEE Vehicle Power and Propulsion Conference*, pp. 1–6, 2008.

[53] D. Krajzewicz, J. Erdmann, M. Behrisch, and L. Bieker, "Recent Development and Applications of SUMO - Simulation of Urban MObility," *International Journal On Advances in Systems and Measurements*, vol. 5, no. 3&4, pp. 128–138, 2012.

[54] S. Kullback and R. A. Leibler, "On information and sufficiency," *Ann. Math. Statistics*, vol. 22, pp. 79–86, 1951.

[55] A. Le Rhun, F. Bonnans, G. De Nunzio, T. Leroy, and P. Martinon, "A bi-level energy management strategy for HEVs under probabilistic traffic conditions," working paper or preprint, 2019, URL: https://hal.inria.fr/hal-02278777.

[56] A. Le Rhun, F. Bonnans, G. De Nunzio, T. Leroy, and P. Martinon, "A Stochastic Data-Based Traffic Model Applied to Vehicles Energy Consumption Estimation," *IEEE Transactions on Intelligent Transportation Systems*, Published online, 2019.

[57] A. Le Rhun, F. Bonnans, G. De Nunzio, T. Leroy, and P. Martinon, "An Eco-routing algorithm for HEVs under traffic conditions," working paper or preprint, Nov. 2019, URL: https://hal.inria.fr/hal-02356277.

[58] D. Lefebvre and M. David, "Dynamic clustering and strong patterns recognition: new tools in automatic classification," *Canadian Journal of Earth Sciences*, vol. 14, no. 10, pp. 2232–2245, 1977.

[59] M. J. Lighthill and G. B. Whitham, "On kinematic waves. II. A theory of traffic flow on long crowded roads," *Proceedings of the Royal Society of London. Series A, Mathematical and Physical Sciences*, pp. 317–345, JSTOR, 1955.

[60] C.-C. Lin, H. Peng, J. Grizzle, and J.-M. Kang, "Power management strategy for a parallel hybrid electric truck," *Control Systems Technology, IEEE Transactions on*, vol. 11, pp. 839 –849, Dec. 2003.

[61] R. Liu and X. Zhu, "Statistical Characteristics of Driver Accelerating Behavior and Its Probability Model," *arXiv:1907.01747*, 2019.

[62] J. MacQueen, "Some methods for classification and analysis of multivariate observations," *Proc. Fifth Berkeley Sympos. Math. Statist. and Probability (Berkeley, Calif., 1965/66)*, Vol. I: Statistics, pp. 281–297, Univ. California Press, Berkeley, Calif., 1967.

[63] A. A. Malikopoulos, "Supervisory Power Management Control Algorithms for Hybrid Electric Vehicles: A Survey," *IEEE Transactions on Intelligent Transportation Systems*, vol. 15, no. 5, pp. 1869–1885, 2014.

[64] A. Martelli, "On the complexity of admissible search algorithms," *Artificial Intelligence*, vol. 8, no. 1, pp. 1–13, Elsevier, 1977.

[65] C. M. Martínez, X. Hu, D. Cao, E. Velenis, B. Gao, and M. Wellers, "Energy Management in Plug-in Hybrid Electric Vehicles: Recent Progress and a Connected Vehicles Perspective," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 6, 2017.

[66] G. Monge, *Mémoire sur la théorie des déblais et des remblais*, De l'Imprimerie Royale, 1781.

[67] M. Montazeri-Gh and M. Mahmoodi-K, "Optimized predictive energy management of plug-in hybrid electric vehicle based on traffic condition," *Journal of Cleaner Production*, vol. 139, pp. 935–948, Elsevier Ltd, 2016.

[68] S. J. Moura, H. K. Fathy, D. S. Callaway, and J. L. Stein, "A Stochastic Optimal Control Approach for Power Management in Plug-In Hybrid Electric Vehicles," *IEEE Transactions on Control Systems Technology*, vol. 19, no. 3, pp. 545–555, IEEE, 2011.

[69] C. Musardo, G. Rizzoni, and B. Staccia, "A-ECMS: An Adaptive Algorithm for Hybrid Electric Vehicle Energy Management," *Proceedings of the 44th IEEE Conference on Decision and Control*, pp. 1816–1823, 2005.

[70] N. J. Nilsson, *Principles of artificial intelligence*, Tioga Publishing Co., Palo Alto, Calif., 1980, pp. xv+476, ISBN: 0-935382-01-1.

[71] F. Österreicher and I. Vajda, "A new class of metric divergences on probability spaces and its applicability in statistics," *Annals of the Institute of Statistical Mathematics*, vol. 55, no. 3, pp. 639–653, 2003.

[72] G. Paganelli, S. Delprat, T.-M. Guerra, J. Rimaux, and J.-J. Santin, "Equivalent consumption minimization strategy for parallel hybrid powertrains," *Vehicular Technology Conference. IEEE 55th Vehicular Technology Conference. VTC Spring 2002 (Cat. No.02CH37367)*, vol. 4, 2076–2081 vol.4, 2002.

[73] A. Panday and H. O. Bansal, "A review of optimal energy management strategies for hybrid electric vehicle," *International Journal of Vehicular Technology*, vol. 2014, Hindawi, 2014.

[74] S. Panwai and H. Dia, "Comparative evaluation of microscopic car-following behavior," *IEEE Transactions on Intelligent Transportation Systems*, vol. 6, no. 3, pp. 314–325, 2005.

[75] P. Pisu and G. Rizzoni, "A Comparative Study Of Supervisory Control Strategies for Hybrid Electric Vehicles," *IEEE Transactions on Control Systems Technology*, vol. 15, no. 3, pp. 506–518, 2007.

[76] J. Pu and C. Yin, "Optimal control of fuel economy in parallel hybrid electric vehicles," *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering*, vol. 221, no. 9, pp. 1097–1106, 2007.

[77] *PWC Spotlight on Automotive*, URL: `https://www.pwc.de/de/automobilindustrie/assets/semiconductor_survey_interactive.pdf`.

[78] J. Rabin, G. Peyré, J. Delon, and M. Bernot, "Wasserstein barycenter and its application to texture mixing," *International Conference on Scale Space and Variational Methods in Computer Vision*, pp. 435–446, 2011.

[79] N. L. Roux, M. Schmidt, and F. R. Bach, "A Stochastic Gradient Method with an Exponential Convergence _Rate for Finite Training Sets," *Advances in Neural Information Processing Systems 25*, pp. 2663–2671, Curran Associates, Inc., 2012.

[80] M. Salman, N. Schouten, and N. Kheir, "Control Strategies for Parallel Hybrid Vehicles," 524 –528 vol.1, Oct. 2000.

[81] A. Sinha, P. Malo, and K. Deb, "A Review on Bilevel Optimization: From Classical to Evolutionary Approaches and Applications," *IEEE Transactions on Evolutionary Computation*, vol. 22, no. 2, pp. 276–295, 2018.

[82] R. Sinkhorn, "A relationship between arbitrary positive matrices and doubly stochastic matrices," *The annals of mathematical statistics*, vol. 35, no. 2, pp. 876–879, JSTOR, 1964.

[83] D. Sperling and D. Gordon, *Two billion cars: driving toward sustainability*, Oxford University Press, 2010.

[84] C. Sun, F. Sun, X. Hu, J. K. Hedrick, and S. Moura, "Integrating Traffic Velocity Data into Predictive Energy Management of Plug-in Hybrid Electric Vehicles," *American Control Conference (ACC)*, 2015.

[85] C. Sun, X. Hu, S. J. Moura, and F. Sun, "Velocity Predictors for Predictive Energy Management in Hybrid Electric Vehicles," *IEEE Transactions on Control Systems Technology*, vol. 23, no. 3, pp. 1197–1204, IEEE, 2015.

[86] O. Sundstrom, P. Soltic, and L. Guzzella, "A Transmission-Actuated Energy-Management Strategy," *IEEE Transactions on Vehicular Technology*, vol. 59, no. 1, pp. 84–92, 2010.

[87] E. Tate, J. Grizzle, and H. Peng, "Shortest path stochastic control for hybrid electric vehicles," *International Journal of Robust and Nonlinear Control*, vol. 18, Sept. 2008.

[88] M. Treiber and A. Kesting, "Traffic flow dynamics," *Traffic Flow Dynamics: Data, Models and Simulation, Springer-Verlag Berlin Heidelberg*, Springer, 2013.

[89] L. N. Vasershtein, "Markov processes over denumerable products of spaces describing large system of automata," *Problemy Peredaci Informacii*, vol. 5, pp. 64–72, 1969.

[90] I. D. Vlieger, D. D. Keukeleere, and J. Kretzschmar, "Environmental effects of driving behaviour and congestion related to passenger cars," *Atmospheric Environment*, vol. 34, no. 27, pp. 4649 –4655, 2000.

[91] Y. Wang, W. Wang, C. Xiang, and X. Wang, "PMP-based equivalent fuel consumption optimization for power distribution of power-split HEVs," *2018 Chinese Control And Decision Conference (CCDC)*, pp. 3445–3450, 2018.

[92] G. Wu, K. Boriboonsomsin, M. J. Barth, and S. Member, "Development and Evaluation of an Intelligent Energy-Management Strategy for Plug-in Hybrid Electric Vehicles," *IEEE Transactions on Intelligent Transportation Systems*, vol. 15, no. 3, pp. 1091–1100, 2014.

[93] B. Yang, J. Dai, C. Guo, C. S. Jensen, and J. Hu, "PACE: a PAth-CEntric paradigm for stochastic path finding," *The VLDB Journal*, vol. 27, no. 2, pp. 153–178, 2018.

[94] S. K. Zegeye, B De Schutter, J Hellendoorn, E. A. Breunesse, and A Hegyi, "Integrated Macroscopic Traffic Flow, Emission, and Fuel Consumption Model for Control Purposes," *Transportation Research Part C*, vol. 31, pp. 158–171, 2013.

université PARIS-SACLAY | École doctorale de mathématiques Hadamard (EDMH)

Titre : Contrôle Optimal Stochastique pour la Gestion Énergétique des Véhicules Hybrides Électriques sous Contraintes de Trafic

Mots Clefs : Controle, Optimisation, Véhicules hybrides, Trafic

Résumé : Cette thèse aborde la conception d'un Système de Gestion Énergétique (EMS), prenant en compte les contraintes de trafic, pour un véhicule hybride électrique. Actuellement, les EMS sont habituellement classés en deux catégories : ceux proposant une architecture en temps réel cherchant un optimum local, et ceux qui recherchent un optimum global, plus coûteux en temps de calcul et donc plus approprié à un usage hors ligne. Cette thèse repose sur le fait que la consommation énergétique peut être modélisée précisément à l'aide de distributions de probabilité sur la vitesse et l'accélération. Dans le but de réduire la taille des données, une classification est proposée, basée sur la distance de Wasserstein, les barycentres des classes pouvant être calculés grâce aux itérations de Sinkhorn ou la méthode du Gradient Stochastique Alterné. Cette modélisation du trafic permet à une optimisation hors ligne de déterminer le contrôle optimal (le couple du moteur électrique) qui minimise la consommation de carburant du véhicule hybride sur un segment routier. Un algorithme bi-niveau tire avantage de cette information afin d'optimiser la consommation sur l'ensemble du trajet. Le niveau supérieur d'optimisation étant déterministe, il est suffisamment rapide pour une implémentation en temps réel. La pertinence du modèle de trafic et de la méthode bi-niveau est illustrée à l'aide de données trafic générées par un simulateur, mais aussi grâce à des données réelles collectées prés de Lyon (France). Enfin, une extension de la méthode bi-niveau au problème d'éco-routage est présentée, utilisant un graphe augmenté pour déterminer l'état de charge lors du chemin optimal.

Title : Stochastic Optimal Control for the Energy Management of Hybrid Electric Vehicles under Traffic Constraints

Keys words : Stochastic Optimal Control, Hybrid Vehicles

Abstract : The focus of this PhD thesis is to design an optimal Energy Management System (EMS) for a Hybrid Electric Vehicle (HEV) following traffic constraints. In the current state of the art, EMS are typically divided between real-time designs relying on local optimization methods, and global optimization that is only suitable for off-line use due to computational constraints. The starting point of the thesis is that in terms of energy consumption, the stochastic aspect of the traffic conditions can be accurately modelled thanks to (speed,acceleration) probability distributions. In order to reduce the data size of the model, we use clustering techniques based on the Wasserstein distance, the corresponding barycenters being computed by either a Sinkhorn or Stochastic Alternate Gradient method. Thanks to this stochastic traffic model, an off-line optimization can be performed to determine the optimal control (electric motor torque) that minimizes the fuel consumption of the HEV over a certain road segment. Then, a bi-level algorithm takes advantage of this information to optimize the consumption over a whole travel, the upper level optimization being deterministic and therefore fast enough for real-time implementation. We illustrate the relevance of the traffic model and the bi-level optimization, using both traffic data generated by a simulator, as well as some actual traffic data recorded near Lyon (France). Finally, we investigate the extension of the bi-level algorithm to the eco-routing problem, using an augmented graph to track the state of charge information over the road network.

Université Paris-Saclay
Espace Technologique / Immeuble Discovery
Route de l'Orme aux Merisiers RD 128 / 91190 Saint-Aubin, France