



# Optimisation de Lois de Gestion Énergétiques des Véhicules Hybrides

Giovanni Granato

## ► To cite this version:

Giovanni Granato. Optimisation de Lois de Gestion Énergétiques des Véhicules Hybrides. Optimization and Control [math.OC]. Ecole Polytechnique X, 2012. English. NNT: . tel-00788160

**HAL Id: tel-00788160**

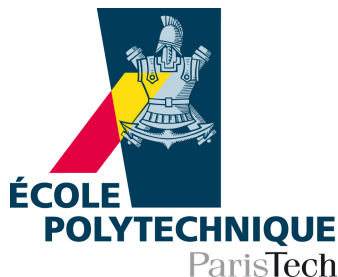
**<https://pastel.hal.science/tel-00788160>**

Submitted on 14 Feb 2013

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

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

ÉCOLE DOCTORALE DE L'ÉCOLE POLYTECHNIQUE



# Thèse

présentée en vue de l'obtention du grade de  
Docteur de l'École Polytechnique

Spécialité : MATHÉMATIQUES APPLIQUÉES

par

**Giovanni Bianchi Granato**

---

## Optimisation de Lois de Gestion Énergétiques des Véhicules Hybrides

---

Thèse soutenue publiquement le 10 Décembre 2012 devant le jury composé de :

M <sup>me</sup> .	HASNAA ZIDANI	Directrice de thèse
M.	FRÉDÉRIC BONNANS	Co-directeur de thèse
M.	YANN CHAMAILLARD	Rapporteur
M.	MAURIZIO FALCONE	Rapporteur
M.	RENÉ HENRION	Rapporteur
M.	PIERRE CARPENTIER	Examineur
M.	AHMED KETFI-CHERIF	Examineur
M.	MARC SCHOENAUER	Examineur





This document was written in  $\text{\LaTeX}$  using the TeXworks working environment.  
Original figures were created using the PGF and TikZ graphic packages.  
Simulations were performed in Matlab, developed by Mathworks.  
Matlab figures were transformed into scalable vector graphics using the `plot2svg` script from Juerg Schwizer.





# Remerciements

*"All the world's a stage, And all the men and women merely players [...]"*<sup>1</sup>

All too often we are quick to cast to ourselves the protagonist role on the epic rhapsody of our lives, neglecting other people to a mere supporting role where they can better enhance the dimension of our own importance. During the time where I worked on my thesis, I came to realize that this over-self-appreciation is a misrepresentation of an utterly different reality. I can only begin to imagine the multitude of people that, with their direct or indirect work and dedication, helped me complete this project. The next words try to modestly convey the expression of my gratitude to those whose contribution was more visible to me.

First and foremost, I would like to thank Professors Yann Chamaillard, Maurizio Falcone and René Henrion, for having accepted to judge and comment on my work. Your technical expertise and experience provided an extremely valuable feedback that served both to enlarge the perspectives of this project and to motivate me to maintain a high standard of work quality: merci, grazie, danke.

Hasnaa, Kamal, Frédéric et Ahmed: trop peu d'espace ici pour dénombrer les nombreuses leçons de mathématiques, d'ingénierie et de vie que vous m'avez apprises. Sachez que je vous tiens en la plus haute estime. Chaque moment que nous avons passé ensemble a fait de moi une meilleure personne que je n'étais il y a trois ans. Merci. Claudia, certainement mon travail n'aurait pas la qualité qu'il a eue sans ton soutien, ta sympathie et tes conseils techniques, toujours très précis et profonds, je souligne. Je t'en remercie. Et aussi, je te remercie d'avoir rendu mon séjour à Rio inoubliable – ton pisco sour, je m'en souviens toujours !

J'ai eu aussi le privilège d'avoir derrière moi une grande entreprise, Renault. Et par grande entreprise, je me réfère aux grands professionnels qui la composent. J'oserai en citer quelques-uns: Maxime, Guillermo, Alexandra, Michel, Pedro, Ana, Cédric (les deux !) Philipe, Hamid, Sébastien, Guillaume, Stéphane, Karima, Ludovic, Sophie-Anne, Jean-Marie, Malek, Laurent. Un grand merci à vous.

Cette thèse a aussi privilégié d'un environnement scientifique stimulant, octroyé, sans doute, par les chercheurs et futurs chercheurs de l'UMA. Les discussions au coin café, toujours, d'ailleurs, arrosées avec du très bon café, ont été tout aussi importantes pour mon épanouissement académique que les plusieurs discussions techniques. Merci beaucoup à vous.

Entre ces deux grands environnements de travail, un petit refuge a occupé une place très ample, composé de rien que de quatre bureaux, des armoires et une immense camaraderie. Nicolas, Lucas et Nicolas, un grand merci ces inoubliables moments partagés, des rires, les débats, le blues, le jazz, le reggae, Bob Dylan, Rolling Stones, Brassens, Brel, happy hours, rappels de topologie et d'analyse (jamais mon fort !). Rien ne serait le même sans vous.

E, fora dessas longas (!) horas de trabalho, devo agradecer também meus velhos camaradas : Guilherme, Gustavo, Aion, JP, Jong, Luciana, Renné ; os (um pouco) mais velhos, Fabi, Manoela, Ângela ; os menos velhos, Daniel, Aron, Gustavo, Louzada, Xandy, Mila, Sabetta, Marini, Julian et Gabriel, assim que as outras denominações de camaradas que se tornaram muito numerosas para que sejam citadas aqui. Espero que cada um saiba a parte que lhes é de direito nesses agradecimentos. Um grande abraço !

Miguel, Inês, Danilo, Arthur. Cada fruto que pude colher até hoje foi graças ao imenso investimento que vocês em mim depositaram. Sinto que nada do que vocês fizeram por mim poderia ter

---

<sup>1</sup> As You Like It - William Shakespeare.

sido melhor. Espero honestamente que eu esteja a altura do que vocês sonhavam para mim. Muito obrigado.

Je ne peux éviter de remercier ici toutes celles qui m'ont rendu la vie plus joyeuse et facile pendant ces dernières années. Puisque je ne peux faire mieux que Brassens, je cite simplement *Les Passantes*.

To all others not explicitly mentioned in this page, I thank you.

Know that, every day, I humbly try to express my gratitude by always putting my best effort into everything I do, so to, perhaps, directly or indirectly, give you as much as you gave me.

---

# CONTENTS

---

<b>1</b>	<b>Models for Hybrid Vehicles and Power Management</b>	<b>9</b>
1.1	Introduction . . . . .	11
1.2	Hybrid Dynamical Systems . . . . .	11
1.3	Optimal Control of Hybrid Systems . . . . .	16
1.4	Hybrid Vehicle Model . . . . .	19
1.4.1	Newton's Equation . . . . .	20
1.4.2	Gearbox, Electric Motor and AC/DC Converter . . . . .	21
1.4.3	Range Extender . . . . .	22
1.4.4	Battery . . . . .	23
1.5	Navigation System . . . . .	24
1.6	Optimal Control and Power Management Systems . . . . .	26
1.6.1	Power Management of Hybrid Vehicles . . . . .	27
1.6.2	Optimal Power Management . . . . .	32
<b>2</b>	<b>Optimal Stochastic Power Management</b>	<b>35</b>
2.1	Introduction . . . . .	37
2.2	Stochastic Power Management . . . . .	37

2.3	Deterministic Dynamic Programming . . . . .	40
2.4	Stochastic Dynamic Programming . . . . .	44
2.5	Simulation Results . . . . .	48
2.5.1	Deterministic Controllers . . . . .	48
2.5.2	Stochastic Controllers . . . . .	50
2.6	Discussion . . . . .	57
<b>3</b>	<b>Stochastic Dual Dynamic Programming and Power Management</b>	<b>59</b>
3.1	Introduction . . . . .	61
3.2	Stochastic Dual Dynamic Programming Implementation . . . . .	63
3.3	Feedback control synthesis . . . . .	67
3.4	Numerical Simulations . . . . .	68
3.4.1	Linear Stochastic Logistics Model . . . . .	68
3.4.2	Linear Stochastic Hybrid Vehicle Model . . . . .	71
3.5	Hybrid Vehicle Model . . . . .	75
3.5.1	Random Variables . . . . .	75
3.5.2	Decision Variables . . . . .	75
3.5.3	Constraints . . . . .	76
3.5.4	Criterion . . . . .	77
3.6	Model Convexity Assessment . . . . .	78
<b>4</b>	<b>Chance Constrained Power Management Strategies</b>	<b>89</b>
4.1	Introduction . . . . .	91
4.2	Probability Constraints . . . . .	91
4.3	Dynamic Programming . . . . .	95
4.4	Chance-constrained Power Management . . . . .	99
4.5	Discussion . . . . .	103
<b>5</b>	<b>Reachability of Hybrid Systems: Theoretical Study</b>	<b>105</b>

5.1	Introduction . . . . .	107
5.2	Motivation and Problem Settings . . . . .	109
5.2.1	Range Extender Electric Vehicles . . . . .	109
5.2.2	Hybrid Dynamical System . . . . .	110
5.2.3	Reachability of Hybrid Dynamical Systems and Optimal Control Problem . .	113
5.3	Main Results . . . . .	115
5.4	Numerical Analysis . . . . .	123
5.4.1	Numerical Scheme and Convergence . . . . .	123
5.4.2	Numerical Simulations . . . . .	126
<b>6</b>	<b>Reachability of Hybrid Systems: Hybrid Vehicle Application</b>	<b>131</b>
6.1	Introduction . . . . .	133
6.2	Hybrid Vehicle Model . . . . .	134
6.3	Reachability Problem and Autonomy . . . . .	137
6.4	Optimal Control Problem and Dynamic Programming Principle . . . . .	138
6.5	Numerical Simulations . . . . .	143
6.6	Discussion . . . . .	149
<b>7</b>	<b>Stochastic Modeling of Speed Profiles</b>	<b>151</b>
7.1	Introduction . . . . .	153
7.2	Data Set Analysis and Model Selection . . . . .	155
7.3	Consumption Forecast Simulations . . . . .	158
7.4	Markov Chain Transition Probability Estimation . . . . .	160



# Foreword

The purpose of this work is to apply optimal control techniques to enhance the performance of the power management of hybrid vehicles. More precisely, the techniques concerned are *viscosity solutions of Hamilton-Jacobi* (HJ) equations, *level set methods in reachability analysis*, *stochastic dynamic programming* (SDP), *stochastic dual dynamic programming* (SDDP) and *chance constrained optimal control*. These are relatively sophisticated optimal control techniques, presenting a rupture (from the application point of view) of more classical techniques (e.g. dynamic programming, maximum principle and heuristic algorithms) found in the literature.

The expected outcome of the Ph.D. project is to assess the general interest in applying such techniques to the power management of hybrid vehicles. This includes stating the relevant modeling choices, implementing a research-level code<sup>1</sup> of the algorithms for simulations and providing a proper interpretation of the simulations results. This represents a first step towards effectively implementing such techniques in series-manufactured hybrid vehicles. Indeed, algorithms implemented into embedded processors must meet stringent safety and reliability levels. This is a costly decision and thus the expected associated gains must represent a competitive advantage at least offsetting this cost. Therefore, a well documented research in laboratory condition is an essential preliminary step.

The conducted research has **six** main contributions.

- The **first contribution** is the introduction of the framework of *hybrid dynamical systems* in the formulation of the optimal power management control problem. This is a convenient framework that relates naturally to certain characteristics of the hybrid vehicle model, in particular the on and off switch of the auxiliary power source and the requirement that successive switchings on and off of the second power source must be separated by a time interval of 120 seconds. In the continuous-time (deterministic) case, the hybrid optimal control problem leads to a system of quasi-variational inequalities (SQVI). We show the existence and uniqueness of the solution of this SQVI in the appropriate viscosity framework and we propose a class of convergent finite-difference schemes to compute the solution. This line of work represents a novel contribution both to the field of hybrid systems and of the field of power management of hybrid vehicles.
- The **second contribution** is the application of level-set methods to compute the reachable set of a deterministic discrete-time model of the hybrid vehicle. From the reachability analysis of the hybrid vehicle power management problem stems two important applications: synthesizing a controller that maximizes the hybrid vehicle range and synthesizing a controller that reaches a final energy repartition chosen by the driver at the beginning of the trip. The novelty is that our framework replaces the state-constrained formulation of the optimal control problem with an *obstacle* problem. This new formulation is particularly well-suited to tackle the maximum reachability problem of hybrid vehicles. In addition, the proposed algorithm, based on a dynamic programming principle, is more readily implementable and offers a more straightforward interpretation of the strategies than finite-difference schemes, an adequate feature for engineering ends.
- The **third contribution** relates to the application of *stochastic dynamic programming* for the power management of hybrid vehicles seeking to minimize the expected financial cost of a trip. More precisely, the novelty with respect to the literature is the introduction of additional route information in the controller synthesis via a *navigation system*. This allows a finite-horizon formulation of the stochastic optimal control problem, resulting in a distance-dependent controller, i.e., receiving feedback of the vehicle position along a route. Classical approaches consider an infinite-horizon formulation, since no route-specific information is assumed, yielding static controllers, receiving state feedback only.

---

<sup>1</sup>In contrast to a prototype-level and embedded-level code.



- The **fourth contribution** arises when noticing that the dimension of the state is a limiting factor whenever using dynamic programming techniques. To tackle this barrier, we study the potential application of a *stochastic dual dynamic programming* (SDDP) algorithm to the optimal power management problem. Since SDDP technique requires the optimization problem to be convex, we propose a relaxed framework, yielding a convex approximation of the original vehicle model.
- The **fifth contribution** seeks to add flexibility to stochastic controllers by introducing a formulation of the optimal control problem using *chance constraints*. By replacing an almost sure constraint, imposing some remaining charge in the battery at the end of a driving cycle, by a chance constraint, the optimization is made over a larger search space. This means that the control strategy accepts a level of failure within a certain confidence bound. The consequence is that an optimal control can be synthesized in driving conditions that were before inaccessible to the optimization procedure. This work formulates the problem of optimal power management with control constraints as a chance constraint optimal control problem. This optimal control problem is then adapted, resulting in an different, but equivalent, formulation. A dynamic programming principle for the equivalent problem is derived. Finally, it is shown how the dynamic programming principle can be applied to a dynamic programming algorithm for numerically synthesizing the optimal control policies.
- The **sixth contribution** concerns the introduction of a *stochastic model of the vehicle speed profiles* as auto-regressive time series using historical data and exogenous information from a navigation system. These models are used to identify the transition probabilities used in the stochastic algorithms, as well as a simulation platform to test the resulting control policies. This approach ensures that the stochastic model used for optimization purposes is the same as the model used for simulating the stochastic policies.

The Ph.D. research project stems from a collaboration between industrial and academic partners. From the industry side this project was financed and supervised by Renault. This document synthesizes the work developed during a research project of 32 months conducted partly at Renault's research facilities located in Guyancourt, France, and in the research laboratories at the ENSTA ParisTech located in Paris, France.

## Avant-propos

Le but de cette thèse est de fournir d'étudier des méthodes mathématiques et numériques pour optimiser la performance de la gestion de l'énergie des véhicules hybrides. Une partie importante de cette thèse est dédiée à la modélisation du problème et à l'évaluation de l'intérêt d'appliquer, à la gestion de l'énergie des véhicules hybrides, différentes techniques telles : commande de systèmes hybrides, équations de Hamilton-Jacobi (HJ) et solutions de viscosité, analyse d'accessibilité et approche level-set, programmation dynamique stochastique (SDP), programmation dynamique duale (SDDP) et la commande optimale avec contraintes en probabilité.

Si le fil conducteur de la thèse est axé sur une application industrielle, cette thèse a pour souci la rigueur mathématique et l'analyse des simulations numériques. L'étude inclut aussi une contribution théorique dans l'analyse des problèmes de commande optimale des systèmes hybrides avec délai entre les commutations.

Plus précisément, les principales contributions de cette thèse sont :

- Mise en place d'un cadre d'étude adapté aux systèmes dynamiques hybrides pour la modélisation du problème de contrôle optimal de la gestion d'énergie de véhicules dotés de plusieurs sources d'énergie. La formulation mathématique proposée semble adéquate pour étudier les dynamiques continues, qui représentent la loi de l'énergie, et les variables de décisions discrètes. Ce cadre permet aussi de tenir compte de plusieurs autres impératifs physiques, notamment l'exigence d'un délai entre commutations successives.
- L'analyse de l'autonomie maximale des véhicules hybrides conduit à l'étude mathématique d'un problème de commande de systèmes hybrides en présence de contraintes sur l'état et avec un délai imposé entre les commutations. La fonction valeur du problème considérée est caractérisée par un système d'inégalités quasi-variationnelles (SQVI). Les résultats théoriques et numériques obtenus pour ce problème constituent une contribution originale en théorie des systèmes hybrides, théorie de la viscosité et résolution numérique des équations HJB.
- Une étude plus spécifique concernant un problème industriel d'autonomie maximale pour un véhicule hybride. Cette analyse propose un algorithme basé sur le principe de programmation dynamique, plus facilement implémentable sur le problème industriel. Cet algorithme donne la loi optimale pour une gestion énergétique de plus grande autonomie et permet d'obtenir une interprétation simple des stratégies des différents régimes.
- Modélisation du problème de loi de gestion d'énergie en tenant compte des aléas de la navigation. Ici, un modèle stochastique des profils de vitesse du véhicule est proposé. Ce modèle est obtenu à partir des données historiques et des informations exogènes fournis par un système de navigation. Cette étude permet d'identifier les probabilités de transition entre les profils de vitesse et aboutit à une plateforme de simulation pour tester les politiques de contrôle. La gestion d'énergie est alors étudiée par utilisation d'un algorithme de programmation stochastique. Étant donné que la dimension de l'état est une limitation numérique lorsqu'on utilise des techniques de programmation dynamique, nous étudions l'application de la programmation stochastique duale (SDDP) pour un système hybride. Cependant la SDDP nécessitant que le problème d'optimisation considéré soit convexe, nous proposons un cadre adéquat qui donne une approximation convexe du modèle d'origine de véhicules hybrides.
- Etude de robustesse des stratégies optimales. Le modèle considéré ici est en présence de contraintes probabilistes sur l'état final de charges des batteries. Un algorithme de résolution pour un problème de contrôle optimal avec contraintes en probabilité est alors analysé pour le modèle de loi de gestion d'énergie.



---

# INTRODUCTION

---

*“Everything in life is somewhere else, and you get there in a car.”*

**Elwyn Brooks White**

Transportation is a quintessential human necessity. Transportation is a key aspect that allows us to fulfill our basic needs for food, water and shelter, as well as more sophisticated needs like social interaction, economic transactions and intellectual development. The place of transportation in our modern civilization is so pervasive that our very cities are designed and built around the way we move. They are conceived to create a better environment for human locomotion.

A major share of the energy produced by mankind is destined towards transportation<sup>2</sup>. Amongst the many ways mankind has created to convey goods and people around the surface of the earth, road transport is predominant<sup>3</sup>. In particular, individual passenger cars are the greatest single mean of passenger transport in most of the developed countries.

Along with the central role the car has in our modern society, comes an increasingly worrisome consequence: pollutants emissions. An important share of greenhouse gas emissions and particle emissions is due passenger cars. These emissions know a steady increase, despite the advancements in related vehicle technology<sup>4</sup>.

There is a consensus in the scientific community that the sharp increase of greenhouse gas emissions, in particular CO<sub>2</sub>, since the last ten decades or so has an effect on the composition

---

<sup>2</sup>Transportation is responsible for over 27 % of final net energy consumption use in the United States, 24 % in EU-25 and 31 % in EU-27[37, 50].

<sup>3</sup>In EU-27, passenger cars represent over 72 % of total passenger transport [38]. The share of roads in the transport of goods is 45 %. In the United States, passenger cars accounts for 80 % of total passengers kilometers, while roads' share in goods transportation is 30 %.

<sup>4</sup>The *Greenhouse Gases, Regulated Emissions and Energy Use in Transportation* (GREET [54]) is a proposed model used to assess the full impact of new vehicle and fuel technologies on energetic efficiency and emissions. This model considers the life-cycle analysis of transport fuels, commonly referred to as *well-to-wheel*, as well as the vehicle life-cycle, comprising manufacturing, usage and recycling.

of the earth atmosphere significant enough to provoke changes in the overall global climate. The extent of such influence on climate is still a matter of debate. However that may be, governments and consumers around the globe are offering incentives to technological developments that help curb the trend of increasing emissions. One of the technological (and commercial) breakthroughs that holds a lot of potential towards a cleaner planet is at the core of this study: *electrified power train technology*. Its most prominent manifestations are in the form of *electric vehicles* and *hybrid vehicles*.

The motivation for this study is the increased development of electrified automobile power train technology. The interest for this technology arises as an increasingly number of automakers wish to adopt vehicles with an electrification of the power chain as a viable solution for reducing greenhouse gas emissions worldwide to meet stringer regulative legislation and consumers' demand<sup>5</sup>. In one hand, there is an effort of major constructors in order to deploy fully electric vehicles (EVs) as early as possible in some car market segments<sup>6</sup>. In the other hand, hybrid vehicle technology has already a relatively important market penetration so far and knows an ever-increasing demand. Moreover, the technological advancements of the hybrid and electric vehicles components is expect to further increase their attractiveness [1].

Among the many aspects of electrified power train technology lies the question of how the vehicle power flow can be controlled. In other words, the controller must chose the share of each individual power source that together meet the total demanded power. Indeed, the extra energy source present in hybrid vehicles allows a greater degree of freedom that can be cleverly exploited towards a further increase in the vehicle overall energetic efficiency.

The contribution of this work consists on the study of how to control the power flow of the vehicle in *the best possible way*. In other words, this document investigates the optimal control of the power flow, also referred to as the *optimal power management strategy*. The proposed line of study seeks to devise the best control strategies when one wishes to optimize the *vehicle range*, the *energy flexibility* or the *financial cost* of a trip. We describe these points further:

**Optimal vehicle range.** Suppose the vehicle has a fixed initial energy distribution between the different energy storage units available (typically a high-voltage battery and a fuel tank). The driver then wishes to drive as far as he/she can before stopping for a recharge. In this scenario, we study techniques of controlling the power flow so that the vehicle reaches the furthest possible point, i.e., it attains its maximum range.

**Optimal energy flexibility.** Suppose now that the vehicle has a fixed initial energy distribution and the driver wishes to drive to a reachable destination. In this context, we propose a control technique that allows the driver to choose, at the start of the trip, the energy distribution of the vehicle at the destination point.

**Optimal cost.** In this particular case, the driver has a strict preference for the cheapest form of energy. Our investigation proposes a method for synthesizing a controller that maximizes the usage of the cheapest energy source to achieve an inexpensive trip.

---

<sup>5</sup>For an analysis of the European scenario see [2].

<sup>6</sup>See [66] for a briefing on the broad issues related to electric vehicle technology.

The novelty of this study lies in the techniques used to tackle the optimization of the power management. These are *viscosity solutions of Hamilton-Jacobi* (HJ) equations, *level set methods in reachability analysis*, *stochastic dynamic programming* (SDP), *stochastic dual dynamic programming* (SDDP) and *chance constrained optimal control*. Another novelty of our work consists on the introduction of the formalism of *dynamic hybrid systems* the the power management of the hybrid vehicle. This work tackles the optimal power management problem with relatively sophisticated optimal control techniques.

This document is structured in **seven chapters**, described as follows:

1. **First Chapter: Models for Hybrid Vehicles and Power Management.** This chapter presents the necessary technical background for the study of optimal power management of hybrid vehicles. The goal of this chapter is to regroup the details about several modeling choices used in our work. The contents of the chapter are mainly divided into two parts: the mathematical framework of hybrid dynamical systems and the engineering tools related to the hybrid vehicle. Concerning the mathematical framework, the chapter introduces the general hybrid systems framework. The optimal control of hybrid systems relevant to our application is then described. Vis-à-vis the application, we describe the components of the vehicle model and state the optimal power management problem as a hybrid optimal control problem.
2. **Second Chapter: Optimal Stochastic Power Management.** The synthesis of efficient power management strategies for hybrid vehicles accounting for uncertainty in the vehicle speed is at the core of this chapter. After the formulation of the optimal control problem, we derive a *stochastic dynamic programming principle* from which one can obtain an algorithm for the synthesis of optimal strategies. Next it describes the simulation model and numerical results of the performance of different kind of control strategies.
3. **Third Chapter: Stochastic Dual Dynamic Programming and Power Management.** This chapter presents a study on the potential application of a *stochastic dual dynamic programming* (SDDP) algorithm to the optimal power management problem. The chapter starts by presenting the general principles behind the SDDP and proposing an implementation of a SDDP algorithm. In particular, attention is draw to the convexity requirements of the optimization problem. Next, two simple numerical examples are constructed to perform a preliminary study on the algorithm. Finally, the chapter presents a hybrid vehicle model that is convex, therefore being adapted to the utilization of a SDDP. This convex hybrid vehicle model is constructed based upon on a complete empirical hybrid model.
4. **Fourth Chapter: Chance Constrained Power Management Strategies.** In this chapter we introduce a chance constrained optimal control problem that can be used to synthesize more flexible optimal control strategies. The chapter presents general remarks about chance constrained problems and introduces a chance constrained optimization problem in a slightly more general formulation than that of the optimal stochastic power management, presented in Chapter 2. Then, the power management problem formulation is adapted to better allow the study of chance constraints. It presents a different, but equivalent, formulation of the optimization problem that can be solved more readily than the original formulation via a dynamic programming algorithm. Last, it details the dynamic programming principle for

the equivalent formulation in a form that can be readily used for the numerical synthesis of optimal feedback using a dynamic programming algorithm.

5. **Fifth Chapter: Reachability of Hybrid Systems - Theoretical Study.** This chapter contains theoretical results regarding the reachability analysis of hybrid systems. There, the reachability set of a continuous-time hybrid system is characterized by a value function via a level set approach. We show that the value function of a hybrid optimal control problem is the unique solution of a system of quasi-variational inequalities in the viscosity sense. Then, we prove the convergence of a class of numerical schemes that allow the numerical computation of the value function. Finally, one particular scheme is implemented and numerical results are presented.
6. **Sixth Chapter: Reachability of Hybrid Systems - Hybrid Vehicle Application.** This chapter studies of the discrete-time dynamical system and the discrete-time optimal control problem for the reachability analysis of hybrid systems. Here, the focus on the discrete-time modeling of the hybrid system leads, instead of a system of quasi-variational inequalities, to a discrete-time dynamic programming principle which can be used to characterize the value function. More precisely, in this approach, the (discrete-time) reachable set is shown to be the level set of a value function associated to a optimal control problem. This value function is shown to satisfy a dynamic programming principle. Then, we construct an algorithm based on this dynamic programming principle to compute the value function. Once the value function is computed we characterize the reachable set and then use an associated minimum time function to synthesize a feedback control that controls the system to the desired energy state. Lastly results of some numerical simulations evaluating the autonomy both of a toy model and realistic vehicle model are presented.
7. **Seventh Chapter: Stochastic Modeling of Speed Profiles.** This chapter details the construction of a stochastic model of the speed profile for electric vehicles. This is an important step in the construction of an optimization model needed later for the synthesis and simulation of stochastic optimal power management strategies. Additionally, a potential application of a stochastic speed model is the forecast of the vehicle power demand in order to obtain an estimation of the vehicle energy consumption. The speed time series is modeled as an autoregressive (AR) model. Taking into account a navigation system that feeds information about the route environment to the power management system, we parameterize the forecast models to some local route parameters. After the models are identified, their performance with respect to the energy consumption of the vehicle is studied. Also, using the autoregressive models, a probability law of the random variable modeling the vehicle future speed is estimated. A kernel density estimation method is used to identify discrete transition probabilities on a sample space obtained through simulations using the AR models.

# MATHEMATICAL MODELING FOR HYBRID VEHICLES OPTIMAL STOCHASTIC POWER MANAGEMENT

---

*“Essentially, all models are wrong, but some are useful.”*

George Edward Pelham Box

## Contents

---

<b>1.1</b>	<b>Introduction</b>	<b>11</b>
<b>1.2</b>	<b>Hybrid Dynamical Systems</b>	<b>11</b>
<b>1.3</b>	<b>Optimal Control of Hybrid Systems</b>	<b>16</b>
<b>1.4</b>	<b>Hybrid Vehicle Model</b>	<b>19</b>
1.4.1	Newton’s Equation	20
1.4.2	Gearbox, Electric Motor and AC/DC Converter	21
1.4.3	Range Extender	22
1.4.4	Battery	23
<b>1.5</b>	<b>Navigation System</b>	<b>24</b>
<b>1.6</b>	<b>Optimal Control and Power Management Systems</b>	<b>26</b>
1.6.1	Power Management of Hybrid Vehicles	27
1.6.2	Optimal Power Management	32

---





## 1.1 Introduction

This chapter presents the necessary technical background for the study of optimal power management of hybrid vehicles. The goal of this chapter is to regroup the details about several modeling choices used in our work.

The contents of the chapter are mainly divided into two parts: the mathematical framework of *hybrid dynamical systems* and the engineering tools related to the *hybrid vehicle*. We stress that the word *hybrid* appearing twice in the last sentence bears two different meanings. In the first case, the term *hybrid dynamical systems* designates a general mathematical framework for modeling a large class of phenomena. These phenomena are similar in the sense that they are governed by continuous-time and discrete-time processes, thus the name hybrid. In the latter case, the term *hybrid vehicle* refers to a vehicle having at least two on-board energy sources.

The chapter is organized as follows. First, an introduction of the hybrid dynamical systems framework is presented. Second, the main aspects of the optimal control of hybrid systems are analyzed, with special focus on the aspects related to our application. Following the mathematical groundwork we start by introducing the hybrid vehicle model used throughout this study in detail. Next, we discuss the aspects related to the vehicle navigation system. The power management issue and, more precisely, the optimal management of power is presented. We end this chapter by introducing uncertainty in the context of power management and discussing the associated modeling choices.

In the following chapters we present a recall of the main points of the models used to render each chapter self-contained. The reader is addressed to this chapter (as well as bibliographical references) whenever details are required.

The next sections are conceived to render the Mathematical Framework and Engineering Tools self-contained and, whenever possible, independent. We hope by proceeding as such, we can better guide the reader to the content of his/her interest.

## 1.2 Hybrid Dynamical Systems

Hybrid dynamical systems (HDS) are a framework used to describe a set of phenomena governed by interacting continuous-time and discrete-time dynamics. Intuitively, one can think of HDS as being a family of continuous-time processes, usually modeled by ordinary differential equations, one of which is ‘active’ at any given time and an associated discrete-time transition rule between such processes. In the case where this transition rule is controllable, HDS can be seen as systems in which some high-level discrete dynamics allows the controller to choose a particular operation mode. The interest in HDS arises naturally when dealing with many application, since several modern control systems involve some high-level logical decision making process coupled with underlying low-level continuous processes [87, 58, 33]. Some examples are flight control systems, production systems, chemical processes and traffic management systems. The term hybrid stems from the different nature of the evolution of the system dynamics, continuous and discrete.

The general formulation of HDS naturally include room for continuous and discrete controllers. In the literature one finds proposed frameworks for HDS modeling [22, 59, 14]. These frameworks vary in their degree of generality. We present our work inspired by the framework in [22] that contains a rather general description of HDS.

In the continuous-time modeling, a dynamic programming approach for the optimal control of hybrid dynamical systems usually leads to a value function that is a solution (in the viscosity frame-

work) of a *system of quasi-variational inequalities* (SQVI). Work [14, Theorem 4.3] characterizes a continuous value function as the unique solution of a SQVI in a viscosity sense in the particular case where no autonomous transitions are allowed. In the case [90] they introduce a weaker notion of solution, namely, discontinuous viscosity solution. The value function is then showed to be a discontinuous viscosity solution of a rather general SQVI, but it fails to be the unique solution. It is shown, however, that solutions of the SQVI share the same lower and upper envelopes.

A general uniqueness result is difficult mainly because the inequalities in the SQVI are coupled non-locally due the action of discrete transitions. In this case, transversality conditions are imposed to obtain better properties of trajectories near the boundary of the domain [30, 8].

Next, we expose a general description of HDS and stress some modeling points that are more salient in our application. Later, we discuss in more detail the aspects of the optimal control of HDS.

We start by defining a (classical) controlled dynamical system.

**Definition 1.1.** A controlled dynamical system is defined as  $\Sigma := [X, \Gamma, f, U]$  where;

- $X, U$  are topological spaces, called respectively continuous state space and continuous control space.  $U$  is supposed to be compact;
- $\Gamma$  is a topological semigroup with an identity element, usually modeling time;
- $f$  is a transition function  $f : X \times U \times \Gamma \rightarrow X$ , often referred to as continuous dynamic of the system.

Often, we shall consider dynamical systems with a state space  $X \subset \mathbb{R}^{d_x}$ ,  $\Gamma = \mathbb{R}^+$  and  $U \subset \mathbb{R}^{d_u}$ , for  $d_x, d_u \geq 1$ . When given a control trajectory  $u : \Gamma \rightarrow U$ , the dynamic  $f$  steers the state  $y$  from the initial value  $x$ , producing a trajectory  $y : \Gamma \rightarrow X$  satisfying

$$y(0) = x, \quad \dot{y}(t) = f(y(t), u(t), t), \quad t \geq 0. \quad (1.1)$$

Such dynamical systems are referred to as *continuous-time dynamical systems*.

An alternative model for dynamical systems can be constructed by considering  $X \subset \mathbb{R}^{d_x}$ ,  $\Gamma = \mathbb{Z}^+$  and  $U \subset \mathbb{R}^{d_u}$ . In this case, given a control sequence  $\{u_0, u_1, \dots\}$ , and initial data  $x$ , the state trajectory  $\{y_0, y_1, \dots\}$  satisfies

$$y_0 = x, \quad y_{t+1} = f(y_t, u_t, t), \quad t \geq 1. \quad (1.2)$$

We refer to such a construction as *discrete-time dynamical systems*.

We present these two particular formulations because they are both employed in the vehicle application.

Hybrid dynamical systems are defined as following.

**Definition 1.2.** A hybrid dynamical system (HDS) is defined as

$$\Sigma_H := [Q, \Sigma, \mathbb{A}, \mathbb{O}, \mathbb{D}, \mathbb{W}, g], \quad (1.3)$$

where

- Set  $Q$  is a discrete set endowed with a discrete topology. Set  $Q$  is called the discrete space state of the hybrid system. This set contains the discrete states  $q$  of the HDS.

- $\Sigma := \{\Sigma_q\}_{q \in Q}$  is a family of dynamical systems, where each  $\Sigma_q = [X_q, \Gamma_q, f_q, U_q]$  is defined as in Definition 1.1.
- Collection  $\mathbb{A} := \{A_q\}_{q \in Q}$ , where each  $A_q \in X_q$ , is the autonomous transitions set.
- Collection  $\mathbb{O} := \{O_q\}_{q \in Q}$ , where each  $O_q \in X_q$ , is the optional transitions set.
- Collection  $\mathbb{D} := \{D_q\}_{q \in Q}$ , where each  $D_q \in X_q$ , is the transition landing set.
- Collection  $\mathbb{W} := \{W_q\}_{q \in Q}$ , where each  $W_q$ , is called discrete control set. Each  $W_q$  describes the possible transitions from state  $q$  to other states in  $Q$ .
- Denoting  $\mathbb{X} = \{X_q\}_{q \in Q}$ , function  $g : \mathbb{X} \times Q \times \mathbb{W} \rightarrow \mathbb{D} \times Q$  is the discrete transition dynamics, or shortly, discrete dynamics.

**Remark 1.1.** For notation convenience, to denote that the discrete mode  $q$  is active, we may employ an index notation such as  $f_q$  or a function notation as in  $f(q)$ .

**Remark 1.2.** This remark seeks to clarify the nomenclature employed throughout this work. More precisely, the remark is made to stress the distinction between what we mean by continuous, continuous-time, discrete and discrete-time.

When using the framework of HDS, there are two kinds of dynamics: the continuous dynamics and discrete dynamics. In Definition 1.2, what is referred to as continuous dynamics groups the family of classical dynamical systems  $\Sigma$ , in particular the continuous state spaces  $X_q$ , the continuous control spaces  $U_q$  and the continuous dynamics  $f_q$ . We stress here that the continuous dynamics  $f_q$  can be a discrete-time dynamics as in Equation 1.2. Otherwise said, even if  $\Sigma$  consists of discrete-time dynamical systems, its elements are referred to as continuous state, continuous control, continuous trajectory, etc. In particular, even if the trajectory  $y$  and control  $u$  are defined by sequences  $\{y_0, y_1, \dots\}$  and  $\{u_0, u_1, \dots\}$  they are called the continuous trajectory and control, respectively.

The term discrete dynamics refers to the transition function  $g$  and the term discrete state space refers to  $Q$ . Also, the family of discrete control spaces is  $\mathbb{W}$ .

At this point we explain briefly the typical behavior of HDS avoiding technicalities that are addressed later on. The state of the system is at first steered by one of the classical dynamical systems, say,  $q_1 \in Q$ . At a given time  $\tau$ , the system undergoes a transition. Set  $W_{q_1}$  contains the available discrete controls at when discrete state  $q_1$  is active. Given  $w \in W_{q_1}$ , the next discrete mode of operation  $q_2$  and the next state value  $y(\tau^+) \in D_{q_2}$  are then given by the value of  $g(y(\tau), q_1, w)$ . This procedure then can be iterated.

With respect to the nature of the transition, they can be categorized in the following manner. A transition is called a *jump* or *impulse* if there is discontinuity in the state trajectory, i.e.,  $y(\tau) \neq y(\tau^+)$ . Otherwise, if the trajectory remains continuous, the transition is referred to as a *switch*. Transitions can be further classified into *autonomous* or *optional* according to whether the state trajectory is inside an autonomous or optional transition set. Indeed, transition can only occur when the state trajectory reaches one set in the collection  $\mathbb{A} \cup \mathbb{O}$ . If the trajectory reaches  $\mathbb{A}$  it must necessarily undergo a transition. However, if trajectory reaches  $\mathbb{O}$  it is left to the controller to decide whether to perform a mode transition or not.

**Remark 1.3.** Definition 1.2 is indeed very general and encompasses a large class of systems. Notice that topological spaces  $X_q$  may be quite different from each other, as well as control sets  $U_q$ . Defined as such, each mode of operation can model distinct underlying physical processes, such as temperature variation, flow adjustment or concentration control in a solution.

The HDS framework allows for a more specific structure aimed at modeling specific behavior of the physical processes underlying the model. For instance, one may include state constraints or a

time delay between autonomous and/or optional transitions. Also, one may wish to produce outputs associated to the continuous or discrete states and/or transitions.

We briefly expose some classical hypothesis on the elements composing a hybrid system  $\Sigma_H$ . This is intended only to complete the general HDS explanation given in this chapter. Because the technical requirements must be adapted to the specific hybrid system in consideration the hypothesis used in each case are explained and detailed in the corresponding chapters.

One needs to ensure that trajectories are well defined for each mode of operation of the HDS. In other words, for all  $q \in Q$ , the dynamical system  $\Sigma_q$  must be well defined.

**Hypothesis 1.1.** For each  $q \in Q$ , the continuous control is a measurable function  $u : \Gamma_q \rightarrow U_q$  such that

$$u(t) \in U_q \text{ for a.e. } t \in \Gamma_q.$$

**Hypothesis 1.2.** For each  $q \in Q$ ,  $f_q(\cdot, \cdot, \cdot) : X_q \times U_q \times \Gamma \rightarrow X_q$  is continuous and there exists  $L_f > 0$  such that

$$\| f_q(y, u, t) - f_q(y', u, t) \| \leq L_f \| y - y' \|, \quad \| f_q(y, u, t) \| \leq L_f \quad (1.4)$$

for all  $t \in \Gamma_q$ ,  $y, y' \in X_q$  and  $u \in U_q$ .

**Hypothesis 1.3.** There exists  $L_g$  such that  $\| L_g \| > 0$  and

$$\| g(y, q, w) \| \leq L_g, \quad \forall q \in Q, \forall y \in \mathbb{X} \text{ and } \forall w \in \mathbb{W}.$$

Moreover, for all  $q \in Q$  and  $w \in \mathbb{W}$ ,  $g(\cdot, q, w)$  is continuous.

In the case  $\Gamma = \mathbb{R}^+$  of continuous-time dynamical systems, Hypothesis 1.1, 1.2 ensure that the trajectory  $y$  satisfying 1.1 exists, is unique and  $y \in W^{1,1}([0, T])$ , with  $0 < T < \infty$ . If  $\Gamma = \mathbb{Z}^+$ , Hypothesis 1.2 ensures a bounded variation from state  $y_{t+1}$  to  $y_t$ . Hypothesis 1.3 avoids jumping into infinity whenever  $\mathbb{D}$  is not bounded or there are infinite many processes.

The next hypothesis concerns the transition phenomena.

**Hypothesis 1.4.** For all  $q \in Q$ , sets  $A_q, O_q$  and  $D_q$  are closed.

**Hypothesis 1.5.** For all  $q \in Q$ ,  $A_q \cap D_q = \emptyset$ .

**Hypothesis 1.6.** For all  $q \in Q$ ,  $A_q \cap O_q = \emptyset$ .

Hypothesis 1.4 ensures that transition times are well defined. Hypothesis 1.5, 1.6 avoid the merging of several autonomous transitions and autonomous and optional transitions occurring at the same time.

The next hypothesis are known as *transversality conditions*.

**Hypothesis 1.7.** For each  $q \in Q$ , boundary  $\partial X_q$  is smooth. Moreover, denoting  $n(x)$  the unit inward-pointing normal vector at  $x \in \partial X_q$ ,

$$f_q(y, u, t) \cdot n(y) \geq 0, \quad \forall y \in \partial X_q \setminus \partial A_q, \forall t \in \Gamma_q, \forall u \in U_q. \quad (1.5)$$

**Hypothesis 1.8.** For each  $q \in Q$ , boundary  $\partial A_q$  is smooth and there exists  $\alpha > 0$  such that

$$f_q(y, u, t) \cdot n(y) > \alpha, \quad \forall y \in \partial A_q, \forall t \in \Gamma_q, \forall u \in U_q. \quad (1.6)$$

Hypothesis 1.7 forbids the system of leaving space  $X_q$  by continuous evolution. Hypothesis 1.8 ensures that the trajectory never approaches the boundary of sets  $A_q$  tangentially avoiding pathological behavior of the trajectory near the boundary  $\partial A_q$ . Similar transversality conditions on  $\partial O_q$  may also be considered in some systems.

As mentioned, in specific cases additional properties of the HDS are needed, such as convexity properties, which require further hypothesis. They shall be presented whenever required.

We present now the concept of *hybrid control*. The main idea is that the hybrid control groups a continuous control trajectory and a discrete control sequence. Thus, given a hybrid control one may obtain a hybrid trajectory for the HDS.

As mentioned, the *continuous control* is a measurable function  $u$  valued in a compact set  $U_q$  that depends on the mode that is currently active. The *discrete control* is a sequence of transition decisions

$$w = \{(w_1, s_1), \dots, (w_i, s_i), (w_{i+1}, s_{i+1}), \dots\}, \quad (1.7)$$

where each  $s_i \in \Gamma$  and  $w_i \in W_{q(s_i^-)}$ . The sequence of discrete transition decisions  $\{w_i\}_{i>0}$  is associated with a sequence of switching times  $\{s_i\}_{i>0}$  where each decision  $w_i$  is exerted at time  $s_i$ . The set of available discrete decisions, at time  $s$ ,  $W_{q(s)}$ , depends on the active mode of operation at the time  $s$ , denoted by  $q(s)$ . We denote by *hybrid control* the control  $a := (u, w)$  grouping both the continuous control and discrete transition decision sequence.

In the following of the document, we restrict our presentation to *finite* horizon problems. Also, we assume, for sake of simplicity, that all  $\Gamma_q \equiv \Gamma$ .

The behavior of the trajectory of a HDS, summarized above, can be synthesized as following. Fix  $T > 0$  and consider  $\Gamma = \mathbb{R}^+$ . Given a hybrid control  $a \in A$  with  $N$  switch orders and given  $q_0 \in Q$ ,  $x := y(s_0^+) \in X_{q_0}$ , the *hybrid trajectory*, denoted by  $(y(\cdot), q(\cdot))$ , verifies

$$\dot{y}(\tau) = f(y(\tau), u(\tau), \tau, q_{i-1}), \quad \tau \in [s_{i-1}, s_i], \quad (1.8a)$$

$$y(s_i^+) = g|_X(y(s_i^-), q(s_{i-1}), w_i) \quad (1.8b)$$

$$q(s_i) = g|_Q(y(s_i^-), q(s_{i-1}), w_i), \quad q(s_0) = q_0, \quad i = 1, \dots, N. \quad (1.8c)$$

Here,  $g|_X$  and  $g|_Q$  denotes the projections of  $g$  onto  $\mathbb{D}$  and  $Q$  respectively, i.e.,  $g = (g|_X, g|_Q)$ .

In the case  $\Gamma = \mathbb{Z}^+$  the system is

$$y_{\tau+1} = f(y_\tau, u_\tau, \tau, q_{i-1}), \quad \tau \in \{s_{i-1}, s_i\}, \quad (1.9a)$$

$$y_{s_i+1} = g|_X(y_{s_i}, q_{s_{i-1}}, w_i) \quad (1.9b)$$

$$q_{s_i} = g|_Q(y_{s_i}, q_{s_{i-1}}, w_i), \quad q_{s_0} = q_0, \quad i = 1, \dots, N. \quad (1.9c)$$

The hybrid trajectory groups the continuous trajectory  $y$  and the discrete trajectory  $q$ .

The notion of *admissible hybrid control* depends on the conditions that must verify the trajectories of the associated HDS. In a general case, a natural way to define an admissible hybrid control is as follows:

**Definition 1.3.** *Given  $0 < t \in \Gamma$ , a hybrid control  $a = (u, w)$  is said to be admissible if the continuous control verifies 1.1 and the discrete control sequence  $w = \{w_i, s_i\}_{i>0}$  has increasing decision times*

$$0 \equiv s_0 \leq s_1 \leq s_2 \leq \dots \leq s_i \leq s_{i+1} \leq \dots \leq T, \quad (1.10)$$

*and admissible decisions*

$$\forall i > 0, \quad w_i \in W_{q(s_i)}. \quad (1.11)$$

Moreover, given  $q \in Q$ ,  $x \in X_q$  and a continuous control  $u(\cdot)$ , define the following hitting times:

$$\mathcal{T}^A(x, q, u(\cdot)) = \inf \{t \mid y(t) \in A_q\}, \quad (1.12)$$

$$\mathcal{T}^O(x, q, u(\cdot)) = \inf \{t \mid y(t) \in O_q\}, \quad (1.13)$$

where  $y(\cdot)$  satisfies

$$\dot{y}(t) = f_q(y(t), u(t), t)$$

with initial condition  $x$ . Then, the transition times must verify

$$\min \left( \mathcal{T}^A(y(s_i^+), q_i, u(\cdot)), \mathcal{T}^O(y(s_i^+), q_i, u(\cdot)) \right) \leq s_{i+1} - s_i \leq \mathcal{T}^A(y(s_i^+), q_i, u(\cdot)). \quad (1.14)$$

We denote by  $\mathcal{A}$  the set of all admissible hybrid controls.

**Remark 1.4.** One remark immediately that, if an infinite number of transitions is ruled out by the appropriated assumptions, in the finite horizon setting, the discrete transition sequence has a finite number of transitions.

It is of course possible to add other conditions in the definition of admissibility for hybrid controls. Once more, it is the nature of hybrid phenomena underlying the HDS model that dictates what conditions are taken into account in Definition 1.3. *Admissible hybrid trajectories* are defined as hybrid trajectories satisfying Equations (1.8a)-(1.8c) engendered by admissible hybrid controls  $a \in \mathcal{A}$ . In the next section, the search for an optimal hybrid trajectory is made only within the set of admissible hybrid trajectories.

We turn now our attention to the formulation of a hybrid optimal control problem.

### 1.3 Optimal Control of Hybrid Systems

After introducing some basic terminology and notation of hybrid dynamical systems, we focus on the optimal control of the continuous-time hybrid systems. This allows to point out the main features of this class of systems.

The cost functional associated to hybrid systems has also a hybrid-like structure. Generally, one may consider a running cost function whenever a continuous operation of the system is in place and place a penalty on each transition made. In this sense, we introduce a *discrete cost function*  $\rho : \mathbb{X} \times Q \times \mathbb{W} \rightarrow \mathbb{R}^+$  and, a *continuous cost function*  $l : \mathbb{X} \times U \times \Gamma \times Q \rightarrow \mathbb{R}^+$ .

In a similar manner, we present usual hypothesis on functions  $l$  and  $\rho$ .

**Hypothesis 1.9.** *There exists  $L_l > 0$  such that*

$$\| l_q(y, u, t) - l_q(y', u, t) \| \leq L_l \| y - y' \|, \quad \| l_q(y, u, t) \| \leq L_l \quad (1.15)$$

for all  $t \in \Gamma_q$ ,  $y, y' \in X_q$  and  $u \in U_q$ .

**Hypothesis 1.10.** *There exists  $L_\rho, L'_\rho > 0$  such that*

$$L'_\rho < \rho(y, q, w) < L_\rho, \quad \forall y \in \mathbb{X}, \forall q \in Q, \forall w \in \mathbb{W}. \quad (1.16)$$

Hypothesis 1.9, 1.10 are introduced to avoid technical difficulties, such as an infinite number of jumps over a finite time period or several jumps accumulating at a single time instant. They can



be interchanged for other appropriated assumptions. For instance, one may have a zero transition cost if one assumes that two consecutive transitions must be separated by a positive time interval.

Fixing  $0 < T \in \Gamma$  and given initial states  $q \in Q$  and  $x \in X_q$  and an admissible hybrid control  $a \in \mathcal{A}$ , one can define the *total operating cost* by

$$J(0, x, q, a) = \int_0^T l(y(\tau), u(\tau), \tau, q(\tau)) d\tau + \sum_{i=1}^N \rho(y(s_i^-), q(s_{i-1}), w_i). \quad (1.17)$$

This total operating cost is used as a criterion to determine the optimality of a hybrid control. An *optimal hybrid control*  $a^*$  is defined as a control such that

$$J(0, x, q, a^*) \leq J(0, x, q, a) \quad (1.18)$$

for any  $a \in \mathcal{A}$ .

The optimal control of HDS can have two different approaches. The first approach is a dynamic programming approach while the second is a (hybrid) maximum principle approach. This latter method seek to extend the Pontryagin's optimality conditions [68] established for classical dynamical systems to the HDS framework. We can refer the interested reader to [40, 5, 25, 84] for related work in the area. Our optimal control formulation is described in terms of the dynamic programming (DP) principle [11, 12].

The total operation cost functional for a system starting with initial states  $q \in Q$  and  $x \in X_q$  at an arbitrary time  $t \in (0, T] \cap \Gamma$ , given a hybrid control  $a \in \mathcal{A}$  is

$$J(t, x, q, a) = \int_t^T l(y(\tau), u(\tau), \tau, q(\tau)) d\tau + \sum_{i=i^*}^N \rho(y(s_i^-), q(s_{i-1}), w_i). \quad (1.19)$$

where  $(y(\cdot), q(\cdot))$  is solution of (1.8a)-(1.8c) and  $i^* = \inf\{1, \dots, N \mid s_i^* \geq t\}$ .

A central concept in DP approach is the *value function*  $v : \Gamma \times \mathbb{X} \times Q$ . The value function is defined as the optimal cost of operation,

$$v(t, x, q) = \inf_{a \in \mathcal{A}} J(t, x, q, a). \quad (1.20)$$

**Remark 1.5.** As a consequence of the regularity of cost functionals  $l$  and  $\rho$  assumed by Hypothesis 1.1-1.10, the value function is bounded. More specifically, this is a direct consequence of the bounds on  $l$  and  $\rho$  and the finiteness on the number of possible transitions.

**Remark 1.6.** Discontinuities in the value function may arise due possible discontinuous transition costs  $\rho$ . Also, if one not consider transversality conditions on sets in  $\mathbb{O}$ , the value function may in general be discontinuous (see [90, Example 2.12] for an example). In a similar manner, the presence of sets  $\mathbb{A}$  without a transversality condition like Hypothesis 1.8 may also engender a discontinuous value function.

We suggest an approach that consists in the formulation of a optimal hybrid control problem without transversality conditions, whose value function is Lipschitz. The problem is formulated in terms of obstacle functions, following the approach used in [17]. We described the formulation in detail in Chapter 5.

We shall present next a version of the *dynamic programming principle* verified by the value function defined in 1.20. As the objective of this exposition is only to enable a discussion about the main elements around the optimal control of HDS, we voluntarily omit specific technicalities. We direct the reader in the quest for more detail to the references at the end of the chapter.

In the framework constructed in our general presentation it is worth to separate the influence of the (admissible) hybrid control on the trajectory in three mutually exclusive cases:



1. The trajectory hits a set  $A \in \mathbb{A}$ , therefore, requiring the system to perform a transition;
2. The trajectory enters a set  $O \in \mathbb{O}$  and may or may not perform a transition;
3. The trajectory is outside  $\mathbb{A} \cup \mathbb{O}$  and it is influenced only by the continuous control.

From definition (1.20), the value function is the optimal with respect to all admissible hybrid controls. Therefore, an optimal hybrid control must optimize all the three cases above. This implies in saying that the value function must verify, in some sense, the inequalities (considering  $\Gamma = \mathbb{R}^+$ )

1. 
$$v(t, x, q) \leq \rho(x, q, w) + v(t, g(x, q, w)) \quad (1.21)$$

for all  $t > 0$ ,  $\forall x \in A_q$  and  $\forall w \in W_q$ ;

2. 
$$v(t, x, q) \leq \rho(x, q, w) + v(t, g(x, q, w)) \quad (1.22)$$

for all  $t > 0$ ,  $\forall x \in O_q$  and  $\forall w \in W_q$ ;

3. 
$$\partial_t v(t, x, q) \leq l(x, u, t, q) + f(x, u, t, q) \cdot \partial_x v(t, x, q) \quad (1.23)$$

for all  $t > 0$ ,  $\forall x \notin A_q$  and  $\forall u \in U_q$ .

Inequalities (1.21)-(1.23) form a *system of quasi-variational inequalities* (SQVI).

The dynamic programming principle can be seen as a statement linking global optimality with local optimality, in the sense that a global optimal trajectory is everywhere locally optimal. Thus, at some point  $(t, x, q)$ , one of inequalities (1.21)-(1.23) must be tight, depending on the location of  $x$  in the state space  $X_q$ . Of course, one still need to establish the correct framework in which the value function is expected to verify (1.21)-(1.23). These remarks summarize the main ideas behind the dynamic programming principle for the value function of a hybrid optimal control problem.

Let us introduce some usual and helpful notation. Define the *optimal transition operator*  $M$  as

$$(Mv)(t, x, q) = \inf_{w \in W_q} \left\{ \rho(x, q, w) + v(t, g(x, q, w)) \right\}. \quad (1.24)$$

In other words, the action of  $M$  in  $v$  produces an optimal transition.

Now, consider the following system of partial differential equations (PDEs):

$$\max \left( v(t, x, q) - (Mv)(t, x, q), \partial_t v(t, x, q) + H(t, x, q, \partial_x v) \right) = 0 \quad \text{in } \mathbb{A}^c \quad (1.25)$$

$$v(t, x, q) - (Mv)(t, x, q) = 0 \quad \text{on } \partial \mathbb{A}^c, \quad (1.26)$$

where  $\mathbb{A}^c$  and  $\partial \mathbb{A}$  denotes the collection of all the complementary sets and boundaries of sets in  $\mathbb{A}$ , respectively and the *Hamiltonian*  $H$  is defined as

$$H(t, x, q, p) = \sup_{u \in U_q} \left\{ -l(x, u, t, q) - f(x, u, t, q) \cdot p \right\}. \quad (1.27)$$

The system (1.25)-(1.26) arising from the SQVI (1.21)-(1.23) can be seen as a *system of Hamilton-Jacobi* (HJ) equations, coupled by a non-local operator serving as a variable boundary condition on the domain  $\mathbb{A}^c$ . It is a system of PDEs because one has one couple of equations (1.25)-(1.26) for each  $q \in Q$ . Since (1.25)-(1.26) play the role of HJ equations, one can use the usual procedure of characterizing the value function via finding the solution of the HJ equations in the viscosity sense [31]. This approach requires uniqueness for the solution of (1.25)-(1.26), usually obtained in the form of a comparison principle. Once a uniqueness result is established, an appropriate numerical scheme allows the computation of the value function by solving a discretization of (1.25)-(1.26).

## 1.4 Hybrid Vehicle Model

This section presents in some detail the vehicle model used throughout the document.

Broadly speaking, *hybrid vehicles* are vehicles that group at least two energy sources whose produced power is directed to move the vehicle. According to [63], a hybrid vehicle means “a vehicle propelled by a combination of an electric motor and an internal combustion engine or other power source and components thereof.” A more general definition is given by [39]: “A hybrid vehicle is a vehicle equipped with at least two energy sources for traction purpose.” Therefore, although one lacks a general precise definition of hybrid vehicles, there is not much loss using the intuitive idea of a vehicle two energy sources collaborating to the movement of the vehicle.

At this date, there are several types of power-train architectures varying on how the electric and mechanic power flows are linked together to produce movement of the vehicle and the dimensions of the specific power-train components, such as batteries, electric motors and internal combustion engines. There is vast documentation on the several types of hybrid vehicles engineered today on the Internet. As our objective is not to produce a detailed account of the different types of hybrid vehicles, we skip a lengthy exposition on the subject. The interested reader should experience no difficulty in finding adequate material. We direct to [73] for an exposition of hybrid vehicles directed towards the optimal control thereof.

In this section, we focus on one particular hybrid vehicle model, namely, the *range extender electric vehicle* (REEV). The REEV consists mainly of an electric vehicle with a small-dimensioned *internal combustion engine* (ICE). The ICE is connected to a fuel tank and to a generator. The fuel tank acts as a reservoir for the fuel that is consumed by the ICE. The generator transforms the mechanical movement of the ICE into electric current, which can in turn be supplied to the main high-voltage network of the electric vehicle (cf. Figure 1.1).

This type of architecture configures a *series hybrid architecture*. The main point of the configuration is that there is no mechanical link between the vehicle wheels and the ICE (otherwise, the configuration is designated, not surprisingly, as a *parallel hybrid architecture*).

The ensemble of the tank, ICE and generator composes the *range extender* (RE) module.

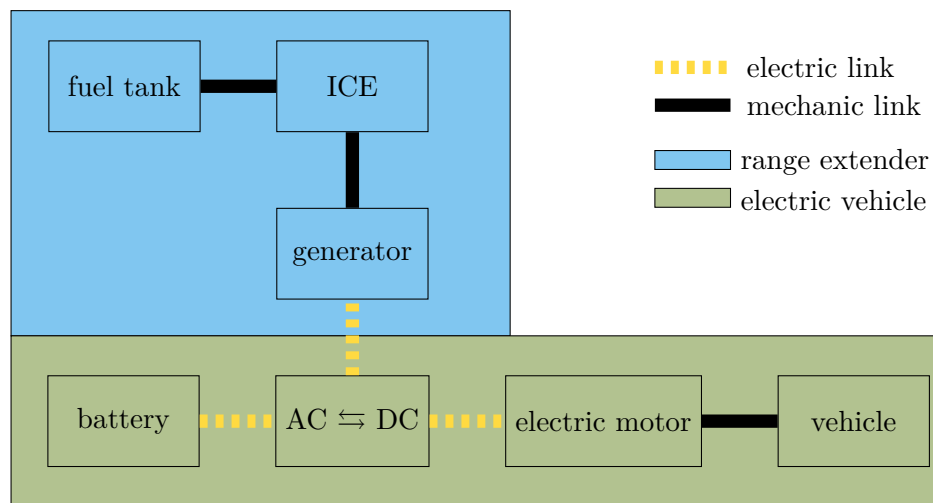


Figure 1.1 – Diagram sketching the series architecture of the range extender electric vehicle model used in this study.

The feature that characterizes the RE module is that *the RE cannot move the vehicle solely by itself*. In other words, the electric current exiting the generator is not enough to ensure that the

vehicle can operate in its nominal mode. Acting alone, the RE can, however, operate the vehicle in a degenerated mode with limited vehicle maximum speed and acceleration.

Besides the extra electric current arriving from the RE module, the REEV works exactly as an *electric vehicle* (EV). The movement of an EV is produced by at least one *electric motor* mechanically linked to the vehicle wheels. The electric motor is controlled by an alternating electric current (AC) fed in the high-voltage network of the vehicle. This electric current is generated by the voltage drop induced in the network by a high-voltage *battery*. In the REEV configuration, the battery acts as the main power source for the vehicle and the RE as a auxiliary power source.

We now describe in more detail the REEV model. In particular, we are interested in how the available energy varies as a function of vehicle parameters and variables. We refer to [44] for a complete exposition and a broader discussion.

### 1.4.1 Newton's Equation

One way of establishing a dynamic model for the REEV is to consider Newton's second law of motion

$$F(t) = \frac{d(mv(t))}{dt} = m\dot{v}. \quad (1.28)$$

Here,  $m$  is the mass of the vehicle (assumed to be constant in time) and  $v(\cdot)$  is the function of the vehicle speed relatively to the ground with respect to time.  $F(t)$  is the resulting force applied to the vehicle as a function of time. We assume that  $F$  and  $v$  have the same direction, horizontal to the ground, and the ground is an inertial reference frame.

At each time, the total force acting upon the vehicle can be decomposed into four terms

$$F = F_g + F_f + F_a + F_d. \quad (1.29)$$

The terms are:

- *Gravitational forces*  $F_g$ . This term is the contribution of terrain inclination that causes a component of the weight of the vehicle to contribute to the forces horizontal to the ground

$$F_g = mg_0 \sin \alpha, \quad (1.30)$$

where  $\alpha$  is the ground inclination and  $g_0$  is the standard gravity.

- *Friction forces*  $F_f$ . Friction forces are caused by the friction between the tire and the ground. They are modeled as

$$F_f = mg_0(C_0 + C_1 v), \quad (1.31)$$

where  $C_0, C_1$  are friction coefficients obtained empirically.

- *Aerodynamic forces*  $F_a$ . These are resistive forces provoked by the air flow surrounding the vehicle. They are described by the expression

$$F_a = \rho C_x S_F (v - v_0)^2. \quad (1.32)$$

In this expression,  $\rho$  is the air density,  $S_F$  is the frontal vehicle surface in contact with the air flow,  $C_x$  is a drag coefficient dependent on the form of the contact surface and  $v_0$  is the wind speed with respect projected onto the vehicle horizontal axis.

- *Dynamic forces*  $F_d$ . This is the force that is generated by the vehicle power-train. The dynamic forces are directed to the vehicle wheels and are given in terms of the torque produced at the wheel:

$$F_d = \frac{T_d}{r_{\text{wheel}}}. \quad (1.33)$$

The wheel radius is denoted by  $r_{\text{wheel}}$ . The torque  $T_d$  has contributions from a *traction* term and a *brake* term, yielding

$$T_d = T_{\text{traction}} + T_{\text{brake}}. \quad (1.34)$$

The term  $T_{\text{traction}} \in \mathbb{R}$  can be positive (for instance, when the vehicle is being propelled forward by consuming fuel), negative (e.g. when the accelerator pedal is not pressed and the vehicle has to cope with the power chain components inertia) or zero (for example, in a negative slope where the weigh of the vehicle compensates the inertia of the power chain components). The term  $T_{\text{brake}} \leq 0$  comes from the torque exercised on the wheels by the braking system and is non-positive.

All constants and coefficients in expressions (1.30)-(1.33) are empirically measured and are therefore consider known and fixed. From (1.28), we write the instant torque from the power-train as a function of the vehicle speed and acceleration under the form

$$T_d = r_{\text{wheel}} (m\dot{v} - F_a(v) - F_f(v) - F_g). \quad (1.35)$$

Now we proceed to link the energy consumption in the battery and in the RE with the torque delivered at the vehicle wheels. To achieve that, we give a dynamic model for each component of the power-chain.

#### 1.4.2 Gearbox, Electric Motor and AC/DC Converter

The electric motor is linked to the vehicle wheels by a gearbox and a differential. The gearbox reduces the rotational speed of the electric motor to be delivered to the differential. The differential allows the left and right wheels to rotate at different speeds, which is required when the vehicle is turning. The electric motor is connected mechanically in the gearbox shaft. As such, they share the same rotational speed  $\omega_{\text{gb}} = \omega_{\text{em}}$ . In our model, there is only one fixed reduction ratio. This can be modeled as  $\omega_{\text{em}} = \omega_{\text{wheel}} R_{\text{gb}}$ , where  $\omega_{\text{em}}$  and  $\omega_{\text{wheel}}$  are the rotational speeds of the electric motor and the vehicle wheels respectively and  $R_{\text{gb}} < 1$  is a fixed reduction ratio. To account for the friction present in the cogwheels that ensure the transmission in the gearbox, we introduce a efficiency term  $\eta_{\text{gb}} \approx 0.95$ . Therefore, to engender a total power output of  $T_{\text{traction}}\omega_{\text{wheel}}$  from the gearbox into the wheels, the total power the electric motor must provide is

$$T_{\text{em}}\omega_{\text{em}} = T_{\text{traction}}\omega_{\text{wheel}}\eta_{\text{gb}}^{-\text{sign}(T_{\text{em}})}, \quad (1.36)$$

yielding

$$T_{\text{traction}} = T_{\text{em}}R_{\text{gb}}\eta_{\text{gb}}^{\text{sign}(T_{\text{em}})}. \quad (1.37)$$

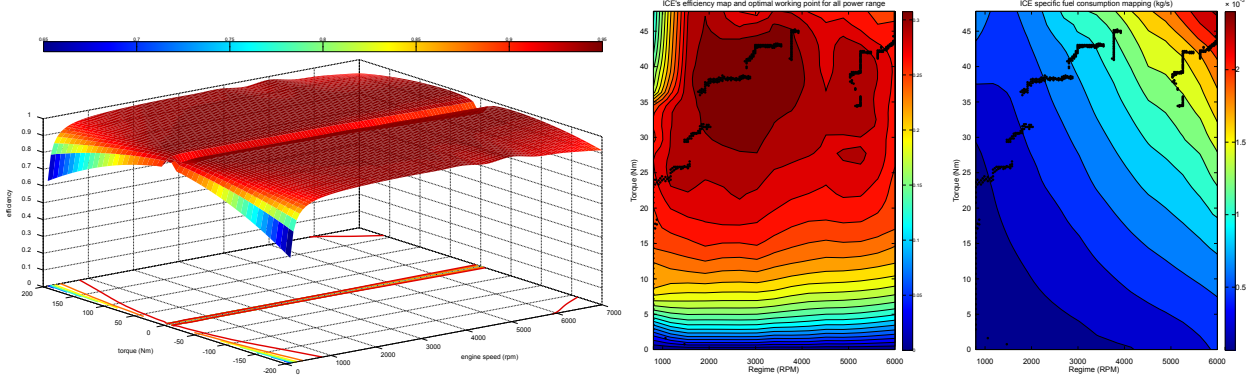
The electric motor transforms the input power, in the form of electric current, into mechanical power, in the form of mechanical torque. The model used in this work is based upon a cartography of the electric motor together with the AC/DC converter. This means that, from our point of view, the electric power entering the AC/DC converter will be transformed in mechanical power exiting the electric motor discounted for the losses in between. The losses are modeled in terms of an

efficiency coefficient  $\eta_{em} < 1$  (cf. Figure ) that depends on the electric motor speed and generated torque. This leads to

$$I_{network}U_{network} = T_{em}\omega_{em}\eta_{em}(T_{em}, \omega_{em})^{-\text{sign}(T_{em})}, \quad (1.38)$$

which, together with (1.37) yields

$$P_{network} = T_{traction}\omega_{wheel}(\eta_{gb}\eta_{em}(T_{traction}, \omega_{wheel}))^{-\text{sign}(T_{traction})}. \quad (1.39)$$



(a) Cartography of  $\eta_{em}$ , the joint efficiency of the electric motor and converter, as a function of the electric motor torque and engine speed.

(b) (left) Cartography of  $\eta_{RE}$ , the joint efficiency of the ICE and generator, as a function of the ICE torque and engine speed. (right) Cartography of the specific consumption  $\dot{m}_{fuel}$  as a function of the ICE torque and engine speed. For a given power, the working points corresponding to the best efficiency are shown in black.

This is the full expression of the electric power on the high-voltage network of the vehicle, given a power demand on the wheel. Otherwise said, when the driver requires some power at the wheel (by controlling the brakes and the accelerator),  $P_{network}$  is the power that is present in the high-voltage network. Remark that  $P_{network}$  may be negative. This means that energy is injected into the high-voltage network, effectively charging the battery.

The next step comes from the observation that this power is the contribution of any power produced in the RE and in the battery.

$$P_{network} = P_{RE} + P_{bat}. \quad (1.40)$$

We discuss this two terms next.

### 1.4.3 Range Extender

The ICE in the RE is mechanically attached to the generator. Thus, for a given mechanical power output of the ICE, there is an electric power output of the generator. The ratio between the mechanical power produced by the ICE and the electric power arriving in the high-voltage network is the RE efficiency. Otherwise put, for a RE efficiency  $\eta_{RE} < 1$ ,

$$P_{RE \rightarrow network} = T_{RE}\omega_{RE}\eta_{RE}(T_{RE}, \omega_{RE}), \quad (1.41)$$

where the power produced in the ICE is  $P_{RE} = T_{RE}\omega_{RE}$ .

In order to produce any power, the ICE needs to consume a certain mass of fuel. The mass variation needed to produce a given power  $P_{RE}$  is also given by a cartography and modeled as an efficiency coefficient  $\eta_{fuel}$ . The efficiency on the fuel consumption is a function of the torque generated by the ICE and the engine speed. To adjust for different units we need to introduce the specific heat of combustion of the fuel  $H_{fuel}$ . Then, the mass variation reads

$$\dot{m}_{fuel}(T_{RE}, \omega_{RE}) = \frac{T_{RE}\omega_{RE}}{H_{fuel}\eta_{fuel}(T_{RE}, \omega_{RE})}. \quad (1.42)$$

### 1.4.4 Battery

The battery is a key element in hybrid vehicle and electric vehicle technology. This component is responsible to transform chemical energy into electric energy that can be used to move the vehicle. Although an electro-chemical modeling of the battery is possible, they reveal themselves in practice computationally intensive and therefore are judged not well suited for the purposes of our application. A very common modeling approach is the equivalent circuit approach. In this approach, the battery is viewed as an equivalent electric circuit whose configuration allows to reproduce to some extent the battery behavior.

A high-voltage battery is composed by an association of *cells*. A cell is the smallest charge storage component in a battery. The chemical composition of a cell and the type of the assemblage used configures a certain type of battery.

In this study we consider a simple battery cell model. This is a steady-state model consists of a voltage source  $U_{oc}$  linked in series with a resistance  $R_{bat}$  as depicted in Figure 1.2.

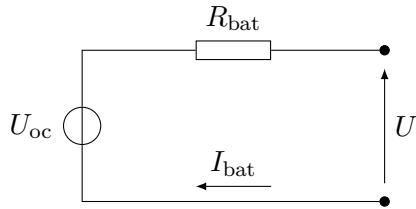


Figure 1.2 – *Equivalent-circuit model of the battery cells.*

This model is characterized by the following quantities:

1. The *open circuit voltage*  $U_{oc}$ . This is the equilibrium potential of the battery. For a given battery, it depends only on the charge. However, it can have different values for different batteries because of the number of cells and the construction methods employed.
2. The *internal resistance*  $R_{bat}$ . This resistance accounts for the resistance in connections, the battery terminals, electrodes and electrolytes. Additionally, this term groups electric resistance arising from the chemical reactions and ionic concentration gradients inside the cell.
3. The *nominal battery maximum charge capacity*  $Q_{bat}$ . This nominal quantity represents the maximum charge that the battery can have obtained after fixed conditions. Indeed, the maximum charge, and thus, maximum energy that the battery is able to store is a variable quantity. Factors such as mean discharge current, discharge time, self-discharge time and overall cells health influence the value of the maximum energy storage capacity of the battery. Therefore, a nominal value for the battery maximum storage capacity is needed for practical reasons.
4. The *state of charge* (SOC)  $s_{bat}$ . The state of charge is an image of the remaining energy in the battery. It is the ratio of the quantity of charge left in the battery and the maximum charge that the battery can contain. However, as mentioned, the maximum possible charge is a very variable quantity. It depends in particular of the charge and discharge cycles used in the battery life. This is the reason why one usually works with a particular value of maximum charge, namely, the nominal battery capacity. The SOC is a normalized by the nominal capacity  $Q_{bat}$ , thus, theoretically ranging from 0 to 1. Although the SOC represents the remainder of the energy in the battery, the link between the two quantities is not straightforward. It is still an open challenge an accurate physical measure of the SOC. In practice, the SOC is estimated using a battery model and measuring other quantities.

From Figure 1.2 one can readily write the relation linking  $U_{oc}$ ,  $R_{bat}$ , the voltage at the battery

terminals  $U$  and the circuit current  $I_{\text{bat}}$ :

$$U_{\text{oc}} = U + R_{\text{bat}} I_{\text{bat}}. \quad (1.43)$$

The power delivered by the battery is  $P_{\text{bat}} = I_{\text{bat}} U$ . Together with (1.43), solving for  $U$  one obtains

$$U = \frac{U_{\text{oc}} + \sqrt{U_{\text{oc}}^2 - 4P_{\text{bat}}R_{\text{bat}}}}{2}, \quad (1.44)$$

where we only keep the greater value (with a  $+$ ) because is the only physically meaningful solution.

An expression for the electric current is then obtained using (1.43):

$$I_{\text{bat}} = \frac{U_{\text{oc}} - \sqrt{U_{\text{oc}}^2 - 4P_{\text{bat}}R_{\text{bat}}}}{2R_{\text{bat}}}. \quad (1.45)$$

Now we establish the link between the instant SOC and the instant electric current  $I_{\text{bat}}$ . By the SOC's definition,

$$\dot{s}_{\text{bat}}(t) = -\frac{I_{\text{bat}}(t)}{Q_{\text{bat}}} \eta_{\text{bat}}(s_{\text{bat}}(t))^{-\text{sign}(I_{\text{bat}}(t))}, \quad (1.46)$$

where the current  $I_{\text{bat}}$  is the instant charge variation in the cells at time  $t$ . Here we choose the convention that  $I_{\text{bat}} > 0$  is a *discharge current*, depleting the charge in the battery. Otherwise,  $I_{\text{bat}} < 0$  is a *charge current*, increasing the SOC level. The factor  $\eta_{\text{bat}}$  is the charge/discharge battery efficiency and depends on whether the current is charging the battery or discharging and depends on the SOC level.

Finally, together with (1.45), one obtains a differential equation for the SOC (omitting the time dependence),

$$\dot{s}_{\text{bat}} = -\frac{U_{\text{oc}}(s_{\text{bat}}) - \sqrt{U_{\text{oc}}^2(s_{\text{bat}}) - 4P_{\text{bat}}R_{\text{bat}}(s_{\text{bat}})}}{2R_{\text{bat}}(s_{\text{bat}})Q_{\text{bat}}} \eta(s_{\text{bat}}), \quad (1.47)$$

where we write  $\eta(s_{\text{bat}}) := \eta_{\text{bat}}(s_{\text{bat}})^{-\text{sign}(s_{\text{bat}})}$  to shorten notation.

In Expression (1.47) we explicit the SOC dependence of the parameters of the battery,  $U_{\text{oc}}$  and  $R_{\text{bat}}$ . The function of these parameters on the SOC may be obtained by further detailed models of the components or empirically. In our study, we use measured cartographies for functions  $U_{\text{oc}}(s_{\text{bat}})$ ,  $R_{\text{bat}}(s_{\text{bat}})$ .

**Remark 1.7.** *We stress here that we consider a relatively simple battery model. In particular, we neglect thermal influence on the battery behavior. Indeed, the temperature is known to affect the electro-chemical reactions in the battery. In a equivalent-circuit modeling, this means that the value of the circuit components depends on the cell temperature, in addition to the SOC. These effects are of considerable magnitude and should to be taken into account in a battery model used for real applications. We refer to [47, 49] for an equivalent-circuit modeling where thermal effects are taken into account.*

## 1.5 Navigation System

A novelty in our approach to power management in hybrid vehicles is the active utilization of a *navigation system* (NAV). It is generally believed that the presence of NAV systems in cars will experiment a growth in the near future. This phenomena may be accentuated by the rising levels of information integration present in today's cars and new future applications. We refer to [85] for an analysis of the typical usage of NAV systems in cars.

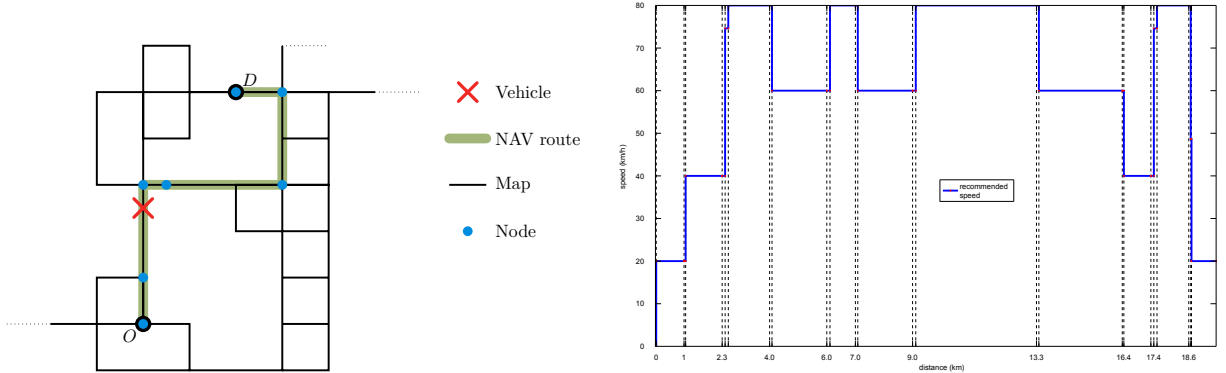


Of the many questions surrounding NAV systems, this work focus on the NAV as a information-providing tool, able to deliver exploitable data to a processor in order to smarter energy usage by the vehicle. The importance of this feature in hybrid and electric vehicles is paramount. For this reason, we briefly present overall information about NAV systems and detail how they are integrated in the power management of hybrid vehicles.

In the light of out application, a navigation system has two main features:

1. Given an origin point and a destination point, the NAV can suggest a route linking the origin and the destination from a given local cartography. A typical utilization scenario is where the origin point is the current geographic location of the vehicle and the driver enters a desired destination in a map. A route is then obtained by the NAV system by optimizing a given criteria, typically the route length, trip time or energy consumption. This implies that the NAV has access to a processor able to perform such computations and a database containing cartographies and related information.
2. The NAV system has access to the current location of the vehicle. Usually this information is in the form of absolute coordinates in the surface of the Earth. This is, however, made more meaningful by placing the vehicle relatively to a local cartography. In particular, given a route suggested by the NAV, the vehicle location can be given relatively to a point in this route. Additionally, the NAV system can also provide the instant speed of the vehicle.

This features are illustrated in Figure 1.3(a).



(a) Figure illustrating the main features of the NAV system for the power management application. The figure presents a local map (black) where it points out the origin point  $O$ , the destination point  $D$ , a route suggested by the NAV (green) and the vehicle current position (red) along the route. The NAV route is divided into nodes and links. The nodes are marked in blue and two consecutive nodes define a link. (b) Example of NAV speed profile. The NAV speed profile presents the NAV recommended cruise speed as a function of the distance from the origin node of the route. The red dots coincide with the route nodes.

We now explain in more detail how the available NAV information is used in the optimization process.

The route suggested by the NAV is composed of *links* and *nodes*. Each link represents a segment of the route. A node is a specific point along the route. A link is defined by an entry node and an exit node. Thus, a route with  $K$  has  $K + 1$  nodes. We shall denote nodes by the index  $k$ , ranging from  $0, \dots, K$ .

The NAV system provides information about some of the attributes of the links composing the route. The two most relevant (and only) attributes used in this study are the *suggested cruise speed* and the *length of the link*. The suggested cruise speed, referred to as *NAV speed* is a driving speed



suggested to the driver according to traffic conditions, route location, weather conditions and alike. They are valued 20km/h, 40km/h, 60km/h, 80km/h, 100km/h or 120km/h. For a given route, the collection of suggested cruise speed in each link is called *NAV speed profile*. Figure 1.3(b) gives an example of such profile.

In this study, the NAV speed profile serve two main purposes:

- In the deterministic setting, we assume the vehicle follows a the NAV speed. Indeed, after adjusting for realistic acceleration and deceleration the speed of the vehicle between links is assumed to be constant.

In this deterministic setting, denoting by  $d_k$  the distance between nodes  $k-1$  and  $k$  and  $\sigma_k > 0$  the (constant) vehicle speed between nodes  $k-1$  and  $k$ , the total time to drive between  $k-1$  and  $k$  is given by  $d_k/\sigma_k$ . This transformation allows us to obtain the vehicle quantities both as a function of the distance from the origin and the time since departure. In particular, the deterministic optimization problems in this work can be equivalently stated in terms of *time* (time since departure) and *space* distance from origin.

- For the stochastic optimization algorithms, the speed of the vehicle at future times is considered to be a random variable. This random variable is modeled as a *random deviation from the NAV speed* (cf. Chapter 7 for details on the stochastic model construction). The rationale behind this model is that the suggested speed contains relevant information about the link's surrounding environment such as speed limit and street type (e.g. highway, freeway, urban). For that reason, although the vehicle will not likely have the exact NAV speed, it will not be far from it, specially if the NAV is actively suggesting a cruise speed.

**Remark 1.8.** *If one wishes to apply a control sequence to the vehicle it is well advised to evaluate each control as a function of the position along the path, rather than as a function of the time since departure. In the case where the vehicle speed is known beforehand, this question is irrelevant and the choice of a space discretization instead of a time discretization is due the nature of the information present in the NAV system. In other words, the RE control are a function of the position of the vehicle on the route, and not the time since departure. If one knows with certainty the vehicle speed, then the two formulations are equivalent. However, the choice of a discretization in space is justified by the possibility of readily transposing the optimal control problem presented in this work if one wishes to consider the vehicle speed as a random variable. Indeed, if the exact vehicle speed is unknown, there is no sense in synthesizing a controller that is applied a certain time after the departure as the vehicle actual position along the path cannot be determined with certainty. In this stochastic framework, there is no certain vehicle speed and one is unable to transform the control as a function of distance from departure into a control as a function of time.*

## 1.6 Optimal Control and Power Management Systems

This section is aimed at presenting our formulation of the optimal control problem for the power management on hybrid vehicles. We structure this presentation in the following form. We first introduce a general optimal control problem in an abstract framework. Then, we define power management for hybrid vehicles. Finally, using the model developed in Section 1.4 we formulate the optimal power management problem in the abstract framework established earlier.

In this section, our focus is the problem statement. To simplify the exposition, this is done in a *deterministic* modeling framework. In the cadre of our work, this is equivalent to say that we consider a *known function of the power demand in the high-voltage network*  $P_{\text{network}}$ . In other words, we consider that, between times 0 and  $t$ , the driver action on the accelerator and brake pedals, i.e. the *driving profile*, is perfectly known. We shall come to that assumption later in the document (Section 2.2).

### 1.6.1 Power Management of Hybrid Vehicles

In this paragraph we define *power management* of hybrid vehicles and state the problem of *optimal power management* as an hybrid optimal control problem.

In the automobile industry, the power management of the vehicle bears a larger meaning than the one used in this work. In the broader sense, the power management of a vehicle deals with *all* aspects of energy storage, distribution and usage within a vehicle. This may include thermal management for cockpit thermal comfort or engine performance, energy routing in the low-voltage network to auxiliaries such as lights, sensors, processors and electronic actuators, water cooling system and management of power-electronics components in the high-voltage network. In this work, whenever the term power management is used, it means *high-level control of the power sources in the vehicle to meet the power demand requested by the driver*. “High-level control” means that we are interested in set-points for the power output of the power sources. In our particular REEV model (cf. Section 1.4), this comes to the power output of the RE  $P_{RE}$  and the power output of the battery  $P_{bat}$  are set. We suppose that they can be set instantly, in our time-scale, by the *low-level* control of the actuators in each component. For instance, in the case of the RE, one can control the mass of the fuel intake in the combustion chamber and the the time that the spark performs the combustion. “Meet the power demand of the driver” means that the driver imposes a power request on the high-voltage network  $P_{network}$  by controlling the brake and accelerator. Furthermore, the relation  $P_{network} = P_{RE \rightarrow network} + P_{bat}$  must hold at all times. Figure 1.3 illustrates the concept of power management.

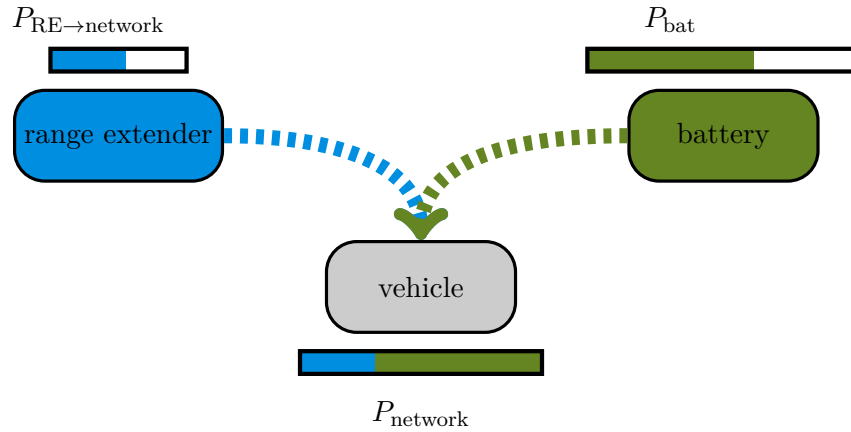


Figure 1.3 – Diagram sketching the principle of a power management strategy. At each time instant, the vehicle imposes a power demand  $P_{network}$  that must be met by the action of the range extender and the battery. The only degree of freedom of the system is  $P_{RE \rightarrow network}$  since, once  $P_{network}$  and  $P_{RE \rightarrow network}$  are fixed,  $P_{bat} = P_{network} - P_{RE \rightarrow network}$  can be determined. A power management strategy consists on the choice of  $P_{RE \rightarrow network}$  at all times.

After giving a precise definition of the concept of power management we proceed to some technical considerations. These shall be consonant with the concepts of the hybrid optimal control theory developed in Section 1.3.

**Continuous control** From the definition of power management, the driver power demand is imposed on the high-voltage network  $P_{network}$ . In an purely electric vehicle, that power demand is met by the battery alone  $P_{bat}$ . In the REEV, the RE can also contribute to meet the requested power by consuming fuel and outputting a power  $P_{RE \rightarrow network}$ . Therefore, since  $P_{network} = P_{RE \rightarrow network} + P_{bat}$  and  $P_{network}$  is imposed, the available degree of the system is the power output from the range extender, namely  $P_{RE \rightarrow network}$ . The power draw from the battery is then a direct consequence of

the remaining power necessary to meet the total demand, i.e.

$$P_{\text{bat}} = P_{\text{network}} - P_{\text{RE} \rightarrow \text{network}}. \quad (1.48)$$

Since the available control is the output power of the RE, we write it as

$$u \equiv P_{\text{RE} \rightarrow \text{network}}. \quad (1.49)$$

This notation allows us to identify the output power of the range extender as the *continuous control* (cf. Remark 1.2).

From (1.41) one sees that the RE power is the product of the torque  $T_{\text{RE}}$  and engine speed  $\omega_{\text{RE}}$ . Since the ICE is not mechanically coupled to the vehicle wheels, we may control both quantities. This means that, for a given power, one can choose freely the values of  $T_{\text{RE}}, \omega_{\text{RE}}$  in the corresponding hyperbole.

**Discrete control** As an additional technical specification, the controller is also allowed to *turn the ICE off and on* during the optimization procedure.

When introducing this flexibility, one is expect to have better (in a sense) control strategies than one would have without it. The switching on and off of the RE might be interesting because of periods where the RE is not requested. Indeed, if the ICE is on, even if it does not deliver any power to the high-voltage network, the engine still needs to consume fuel to keep turning at a predetermined level to avoid stalling. Therefore, if a control strategy finds it best not to use the RE to save fuel, switching it off can prevent an additional, non negligible, useless fuel consumption. The *switch control sequence* of the RE is denoted  $w$ . This on/off control sequence is identified as the *discrete control sequence*, detailed in the hybrid framework in Expression (1.7) and detailed in Definition 1.3.

**Control constraints** We start by describing the constraints on the continuous control. The choice of  $T_{\text{RE}}, \omega_{\text{RE}}$  is subject to technical constraints.

First, the engine speed has fixed upper and lower bounds,

$$\underline{\omega} \leq \omega_{\text{RE}} \leq \omega_1. \quad (1.50)$$

An upper bound is needed in order to prevent damaging the engine. Also to ensure the physical safety of the ICE, a lower bound is needed to avoid vibrations that usually resonate at low engine speed.

Additionally, the engine speed must respect an upper bound that depends on the *vehicle speed*,

$$\omega_{\text{RE}} \leq \omega_2(v). \quad (1.51)$$

This constraint is needed to prevent acoustic nuisance occurring at elevated engine speeds inside the cockpit, in particular for low vehicle speeds.

The bound constraints on the engine speed are then written as

$$\underline{\omega} \leq \omega_{\text{RE}} \leq \bar{\omega}(v), \quad (1.52)$$

where  $\bar{\omega}(v) := \min(\omega_1, \omega_2(v))$ .

The torque delivered by the ICE at the entry of the generator has also lower and upper bounds,

$$0 \leq T_{\text{RE}} \leq \bar{T}(\omega_{\text{RE}}). \quad (1.53)$$

The lower bound ensures that the torque is always non-negative, while the upper bound represents the maximum torque capacity of the ICE and depends on the engine speed.

The constraints just described must be verified at all times by the ICE. The optimal power management problem is formulated in terms of the total power delivered by the RE. This simplification condenses the two degrees of freedom (engine speed and torque) into only one, the ICE output power. Since the model is formulated in terms of the ICE power, the constraints on the ICE power are written in terms of constraints (1.52), (1.53) as

$$0 \leq u \leq \bar{u}(v) := \bar{T}(\omega_{\text{RE}})\bar{\omega}(v)\eta_{\text{RE}}(T_{\text{RE}}, \omega_{\text{RE}}). \quad (1.54)$$

We now describe the constraints on the discrete switching sequence. Let us recall briefly that the switching sequence  $w$  is defined as a sequence of  $N$  switching orders  $w_j$  and  $N$  switching positions  $s_j$ :

$$w = \{(w_1, s_1), \dots, (w_N, s_N)\}. \quad (1.55)$$

The  $s_j$  can be identified with any points of the route. In particular, they can be identified as being the path nodes (cf. Section 1.5).

In the REEV model, only two decision are available: *turning on* the ICE and *turning off* the ICE. Thus, we require

$$w_j \subset W = \{0, 1\}, \quad j = 1, \dots, N, \quad (1.56a)$$

$$s_1 \leq \dots \leq s_N. \quad (1.56b)$$

We now introduce an important condition in our model, referred to as *lag constraint*. The lag constraint imposes a positive *time* interval  $\delta > 0$  between two consecutive switches. In our application, the lag interval imposed between transitions is of  $\delta = 120$  seconds. In the deterministic framework we known the vehicle speed  $v_i$  at each node  $i$  of the route. From the navigation system one can known the distance  $d_i$  between nodes  $i - 1$  and  $i$ . Assuming constant speed between nodes and a discrete-time setting, knowing  $d_i$  and  $v_i$ , the total time spent between nodes  $i - 1$  and  $i$  is  $d_i/v_i$ . The lag condition then requires that between two consecutive switch decisions,  $s_j$  and  $s_j + 1$ , the total time spent is at least 120s. This condition can be written as

$$\sum_{i=1+s_j}^{s_{j+1}} \frac{d_i}{v_i} \geq \delta. \quad (1.57)$$

The concrete interest in this condition is that by limiting the number of start-ups and shutdowns of the ICE, the engine suffers less mechanical wear. The switch limitation imposed by the lag condition contributes to prolonging the lifespan of the ICE.

**Remark 1.9.** *The lag condition (1.57) fits well into the framework of hybrid systems. When modeling the hybrid vehicle as a continuous-time hybrid system in Chapter 5, the lag constraint allows to avoid some technical difficulties concerning the finiteness of transitions.*

**Discrete state variables** The discrete switching control is used to turn on and off the ICE in the RE module. Thus natural quantity here acting as a discrete variable, modeling the mode of operation of the RE, is the *ICE status* or the *ICE state*, denoted  $q$ .

Whenever a switch order  $w_j$  is performed, the ICE state changes according to

$$q_{i+1} = w_j(1 - q_i) + (1 - w_j)q_j. \quad (1.58)$$

The discrete state takes the values  $q = 1$  if the ICE is on and  $q = 0$  if the ICE is off.

**Continuous state variables** In the deterministic setting, the *state* of the system is a two-dimensional vector composed of the SOC and the fuel mass.

The SOC, which is a normalized quantity is restricted to lie in a tighter set than  $[0, 1]$ :

$$0 < \underline{s} \leq s_{\text{bat}} \leq \bar{s} < 1. \quad (1.59)$$

The bounds  $\underline{s}$  and  $\bar{s}$  are safety bounds that restrict charging or depleting the battery too much. Indeed, a charge level above  $\bar{u}$  or below  $\underline{u}$  can permanently damage the cells in the battery, which is undesirable.

The fuel mass is only limited by its existence and the volume of the tank, resulting in the constraint

$$0 \leq m_{\text{fuel}} \leq \bar{m}. \quad (1.60)$$

For notational convenience we shall denote the fuel mass and the SOC as

$$s_{\text{bat}} \equiv y, \quad (1.61)$$

$$m_{\text{fuel}} \equiv r. \quad (1.62)$$

The SOC dynamics (1.47) is rewritten as

$$\dot{y}(t) = f(y(t), u(t), q_i, t). \quad (1.63)$$

They depend on the mode of operation of the RE  $q_i$  active at time  $t$ . Indeed, making the dynamics  $f$  explicit, one obtains (using (1.48))

$$f(y, u, q_i, t) = -\frac{U_{\text{oc}}(y) - \sqrt{U_{\text{oc}}^2(y) - 4R_{\text{bat}}(y)(P_{\text{network}}(t) - uq_i)}}{2R_{\text{bat}}(y)Q_{\text{bat}}} \eta(y). \quad (1.64)$$

Observe the effect of the mode of operation  $q_i$  on the SOC dynamics. If the RE is off,  $q_i = 0$  and the RE power contribution is suppressed. Otherwise, the RE is on,  $q_i = 1$  and the power can be used to decrease the SOC depletion rate accordingly.

The discrete control lag condition formulated in (1.57) is not well adapted to a dynamic programming principle formulation, needed later on. In order to include this lag condition in the optimal control problem in a more suitable form, we introduce a state variable  $\pi$ . Recall that the decision lag conditions implies that new switch orders are not available up to a time  $\delta$  since the last switch. The new variable is constructed such that at a given time  $\tau$ , the value of  $\pi(\tau)$  measures the *time since the last switch*. The idea is to impose constraints on this new state variable and treat them more easily in the dynamic programming principle. Thus, if  $\pi(\tau) < \delta$  all switch decisions are blocked and if, conversely,  $\pi(\tau) \geq \delta$  the system is free to switch. For that reason, this variable can be seen as a switch lock.

Now given a final time horizon  $T > 0$ ,  $\tau \in [0, T]$ , a initial condition  $p \in [0, \delta]$  and a discrete control  $w = \{w_i, s_i\}_{i>0}$ , the switch lock dynamics is defined by

$$\pi(\tau) = \begin{cases} p + \tau & \text{if } \tau < s_1 \\ \inf_{s_i \leq \tau} \tau - s_i & \text{if } \tau \geq s_1 \end{cases} \quad (1.65)$$

Observe that once given a discrete control, the trajectory  $\pi(\cdot)$  can be determined. Then, the lag condition is translated in the point-wise state constraint

$$\pi(s_i^-) \geq \delta \quad (1.66)$$

for all  $s_i$ , where  $s_i^-$  denotes the limit to the left. Notice moreover that  $\pi(s_i^+) = 0$  by construction.

These conditions suffice to define an admissible discrete control set. Therefore, while optimizing with respect to admissible controls, one needs only look within the set of discrete switching sequence that engenders a trajectory  $\pi(\cdot)$  with the appropriate structure (1.66).

When starting-up the ICE, there is an associated fuel consumption. This extra fuel consumption is necessary to counter the inertia in the engine moving parts until a nominal working point is established. This fixed extra consumption  $\rho_0 > 0$  has to be taken into account in the mass dynamics equation. Since the condition  $\pi = 0$  characterizes a switch, the dynamics of the mass consumption can be written as

$$\dot{r}(t) = l(u(t), \pi(t), q_i, t) := -o(u(t), t) - \rho_0 \mathbf{1}_{\{\pi=0\}}(\pi)(1 - q_i). \quad (1.67)$$

where  $q_i$  is the active RE state and  $\mathbf{1}_A(x)$  is the indicator function of event  $x \in A$ . Function  $o(\cdot, \cdot) \geq 0$  is the specific fuel consumption (1.42).

In summary, the state vector is composed of three state variables, the SOC, the fuel mass and the time since last switch, respectively  $y, r, \pi$ . The continuous dynamics of the state vector are, given continuous and discrete controls  $u(\cdot), w$ , and initial conditions  $y^*, r^*, p, q$  given by

$$\dot{y}(t) = f(y(t), u(t), q_i, t), \quad (1.68)$$

$$\dot{r}(t) = l(u(t), \pi(t), q_i, t), \quad (1.69)$$

$$\pi(t) = \begin{cases} p + t & \text{if } t < s_1 \\ \inf_{s_i \leq t} t - s_i & \text{if } t \geq s_1 \end{cases}, \quad (1.70)$$

$$y(0) = y^*, \quad (1.71)$$

$$r(0) = r^*, \quad (1.72)$$

and at switch nodes  $s_j, j = 1, \dots, N$ , the intervening discrete dynamics is

$$q_{i+1} = w_j(1 - q_i) + (1 - w_j)q_i, \quad (1.73)$$

$$q_1 = q. \quad (1.74)$$

**Criterion** The optimal power management can be viewed as optimizing the efficiency with which the on-board energy is used. This amounts to minimizing the energy spent on a trip and hence, to maximizing the quantities related to that energy at the end of a trip, namely, the remaining SOC and fuel. In a deterministic setting, the quantity to be minimized is the energy used by the control strategy, imaged in the fuel and SOC consumption. More precisely, the optimization criterion  $J$  (we recall, to be minimized) is, given  $T$  optimization stages and a hybrid control variables  $a = (u, w)$ ,

$$J(y^*, r^*, p, q; a) = (r^* - r(T)) + \beta(y^* - y(T)). \quad (1.75)$$

Here,  $(y^*, r^*, p)$  denotes a initial value for the state vector  $(y, r, \pi)$ ,  $q$  is an initial RE state and The fixed parameter  $\beta \geq 0$  is a constant factor used for units adjustment.

In this study,  $\beta$  is set as the ratio of the prices of the different energy sources, converting the units of the SOC (typically in % of the total available nominal charge capacity) and the fuel (given, for instance, in liters) to, say, euros. Such conversion yields  $\beta = 0.44 \text{ €/MJ}$  at today estimated fuel and electricity prices<sup>1</sup>.

<sup>1</sup>The factor  $\beta$  is seen as a unit conversion factor. In this sense, it is not a weight factor representing a preference of one energy over another. Even though one can argue that the choice of economic cost is implicitly a kind of preference of one energy over another, this is an objective choice. We choose to peg  $\beta$  to an objective measure (the price of energy) to work with a fixed numerical value. We leave open the debate of whether and how  $\beta$  may represent the *value to the driver* of one type of energy over another. For instance, if the driver seeks the strategy with least pollutants emission, one can set  $\beta = 0$  and if the application requires saving the battery's energy, one can set  $\beta \gg 1$ .

### 1.6.2 Optimal Power Management

The model detailed in the above paragraphs enables us to state the *optimal power management* problem as a hybrid optimal control problem. The objective of such optimization is to minimize the total financial cost of the operation of the range extender on a given trip.

The electric vehicle is a technology that begins to penetrate the auto market. The EV technology is attractive for the client for two main reasons: it is an ecological and economical vehicle (see [53] for an analysis of the energetic impact of electrified power-train technology in the U.S.). The ecological argument is straightforward. An EV has zero (tank-to-wheel) pollutants emissions and produces very little noise. These features contribute to the overall environmental quality, specially in big urban centers, where the EV is expected to reach its biggest market penetration. Economically, the EV has a mid- to long-term advantage over classic vehicles. First, an EV has maintenance costs  $\approx 20\%$  less than fuel powered vehicles. Second, in 2012 prices, electric energy costs are about half of the equivalent fuel generated energy. Since the EV is sold (without the battery) at roughly the same price of vehicles in the same category, the driver can expect a cost reduction related to personal transportation, directly proportional to his/hers millage. The main practical issue that EV technology faces is the relatively low range ( $\approx 200\text{km}$ ). The range extender option is therefore placed to relief the client of some of the anxiety related to the EV autonomy. In the case of an ICE acting as a RE, to maintain the main advantages of the EV technology and reassure the client about the vehicle autonomy, the presence of the RE is required, but it is recommended that its usage be minimal. The optimal power management of REEV is a tool aimed at achieving the goal of minimizing the cost of operation of the RE, while at the same time ensuring that the RE is used whenever is needed.

Vis-à-vis the mathematical formulation, we state the optimal control problem associated to the optimal power management in two ways in this work. More precisely, we formulate a *continuous-time* and a *discrete-time* hybrid optimal control problem.

With the notation of Paragraph 1.6.1, the discrete-time formulation is as follows:

Let  $y^*, r^*, p$  be the initial SOC, the initial available fuel and the time required to the next switch of the RE respectively. Let  $q$  be an initial RE state. Given a route composed of  $K + 1$  nodes, a speed profile  $\sigma_i$  and link lengths  $d_i$ ,  $i = 1, \dots, K$ , find a hybrid control sequence  $a^* = (u, w)$ , where  $u = (u_1, \dots, u_K)$  and  $w = \{(w_1, s_1), \dots, (w_N, s_N)\}$  at switching nodes  $s_j$ , such that, for all  $k = 1, \dots, K$  and  $j = 1, \dots, N$ :

$$a^* \in \arg \min_{a \in A} \left\{ (r^* - r_K) + \beta(y^* - y_K) \right\}, \quad (1.76a)$$

subject to continuous dynamics

$$y_{k+1} = y_k + f(y_k, u_k, q_i, k) \frac{d_k}{\sigma_k}, \quad (1.76b)$$

$$r_{k+1} = r_k - o(u(t), t) \frac{d_k}{\sigma_k} - \rho_0 \mathbf{1}_{\{\pi_k=0\}}(\pi_k)(1 - q_i), \quad (1.76c)$$

$$\pi_k = \begin{cases} \sum_{i=2}^k \frac{d_i}{\sigma_i} & \text{if } 2 \leq k < s_1 \\ \inf_{s_j \leq k} \sum_{i=s_j}^k \frac{d_i}{\sigma_i} - \frac{d_{s_j}}{\sigma_{s_j}} & \text{if } k \geq s_1 \end{cases}, \quad (1.76d)$$

with initial conditions

$$y_1 = y^*, \quad (1.76e)$$

$$r_1 = r^*, \quad (1.76f)$$

$$\pi_1 = p, \quad (1.76g)$$



discrete dynamics

$$q_{i+1} = w_j(1 - q_i) + (1 - w_j)q_i, \quad (1.76h)$$

$$q_1 = q, \quad (1.76i)$$

under control constraints

$$0 \leq u_k \leq \bar{u}, \quad (1.76ja)$$

$$w_j \subset W = \{0, 1\}, \quad (1.76jb)$$

$$s_1 \leq \dots \leq s_N, \quad (1.76jc)$$

and state constraints

$$\underline{s} \leq y_k \leq \bar{s}, \quad (1.76ka)$$

$$0 \leq r_k \leq \bar{m}, \quad (1.76kb)$$

$$\pi_{s_j} \geq \delta, \quad (1.76kc)$$

$$q \in \{0, 1\}. \quad (1.76kd)$$

The continuous-time model is stated considering that the discrete control sequence is composed of switching orders and switching *times*, instead of switching nodes. This simplification provides a sensible simplification in notation without losing any generality (see Remark 1.8). The continuous-time optimal power management problem is then formulated as:

Let  $y^*, r^*, p$  be the initial SOC, the initial available fuel and the time required to the next switch of the RE respectively. Let  $q$  be an initial RE state. Consider a route with total travel time of  $T > 0$  and let  $t \in [0, T]$ . The optimal power management is a hybrid control sequence  $a^* = (u, w)$ , where  $u \in L^\infty([0, T], \mathbb{R})$  and  $w = \{(w_1, s_1), \dots, (w_N, s_N)\}$  at switching times  $s_j$ , such that, for a.e.  $t \in [0, T]$  and  $j = 1, \dots, N$ :

$$a^* \in \arg \min_{a \in A} \left\{ (r^* - r(T)) + \beta(y^* - y(T)) \right\}, \quad (1.12a)$$

subject to continuous dynamics

$$\dot{y}(t) = f(y(t), u(t), q_i, t), \quad (1.12b)$$

$$\dot{r}(t) = l(u(t), \pi(t), q_i, t), \quad (1.12c)$$

$$\pi(t) = \begin{cases} p + t & \text{if } t < s_1 \\ \inf_{s_i \leq t} t - s_i & \text{if } t \geq s_1 \end{cases}, \quad (1.12d)$$

with initial conditions

$$y(0) = y^*, \quad (1.12e)$$

$$r(0) = r^*, \quad (1.12f)$$

discrete dynamics

$$q_{i+1} = w_j(1 - q_i) + (1 - w_j)q_i, \quad (1.12g)$$

$$q_1 = q, \quad (1.12h)$$

under control constraints

$$0 \leq u(t) \leq \bar{u}, \quad (1.12ia)$$

$$w_j \subset W = \{0, 1\}, \quad (1.12ib)$$

$$s_1 \leq \dots \leq s_N, \quad (1.12ic)$$



and state constraints

$$\underline{s} \leq y(t) \leq \bar{s}, \quad (1.12\text{ja})$$

$$0 \leq r(t) \leq \bar{m}, \quad (1.12\text{jb})$$

$$\pi(s_j) \geq \delta, \quad (1.12\text{jc})$$

$$q \in \{0, 1\}. \quad (1.12\text{jd})$$

A hybrid control  $a$  satisfying problem (1.76) or (1.12) is said to be an *optimal power management strategy*. Chapter 2 presents in detail the methods used to obtain an optimal power management strategy.

# OPTIMAL STOCHASTIC POWER MANAGEMENT

---

*“I never appreciated ‘positive heroes’ in literature. They are almost always clichés, copies of copies, until the model is exhausted. I prefer perplexity, doubt, uncertainty, not just because it provides a more ‘productive’ literary raw material, but because that is the way we humans really are.”*

José Saramago

The content of this chapter is an adaptation from the following publication and oral communication:

- Aouchiche K., Bonnans F., Granato, G., and Zidani H. *A stochastic dynamic principle for hybrid systems with execution delay and decision lags*, Decision and Control and European Control Conference (CDC-ECC), 2011 50th IEEE Conference on , pp.6788-6793, 12-15 Dec. 2011, doi: 10.1109/CDC.2011.6161303.
- Aouchiche K., Bonnans F., Granato G. and Zidani H. *Stochastic Optimization of Hybrid Systems with Activation Delay and Decision Lag*, International Conference on Continuous Optimization- ICCOPT, 2010, Santiago, Chile, *oral communication*.

## Contents

---

<b>2.1</b>	<b>Introduction</b>	<b>37</b>
<b>2.2</b>	<b>Stochastic Power Management</b>	<b>37</b>
<b>2.3</b>	<b>Deterministic Dynamic Programming</b>	<b>40</b>
<b>2.4</b>	<b>Stochastic Dynamic Programming</b>	<b>44</b>
<b>2.5</b>	<b>Simulation Results</b>	<b>48</b>
2.5.1	Deterministic Controllers	48
2.5.2	Stochastic Controllers	50
<b>2.6</b>	<b>Discussion</b>	<b>57</b>

---



## 2.1 Introduction

The synthesis of efficient power management strategies for hybrid vehicles accounting for uncertainty in the vehicle speed is at the core of this chapter. The three main steps considered towards fulfilling this goal are: the construction of an *optimization model*, the implementation of an *algorithm* for the controller synthesis and the subsequent utilization of a *simulation model*. Each one of these steps is detailed in the following sections.

We consider a stochastic approach to obtain control strategies that are robust enough to account for the uncertainty of the vehicle speed profile. In this context, the goal becomes to synthesize power management strategies that are *optimal in average* with respect to the efficiency of the utilization of the energy available on-board. Our approach includes the utilization of information from the vehicle navigation system as well as a statistical analysis of previously stored driving data.

The construction of an optimization model consists on a series of approximations of the real system and the real environment. These approximations can be separated into two classes: the approximation of the physical behavior of the components of the vehicle and an approximation for the stochastic process underlying the evolution of the vehicle speed. The physical models used for approximating the physical behavior of the vehicles components is discussed in Chapter 1. The construction of a model for describing the random evolution of the vehicle speed and the associated energy consumption is discussed in Chapter 7.

Once both models are obtained, one can use them to state an *optimal control problem*. The term optimal control problem refers to a formulation of the optimization problem we must solve to achieve the objective of synthesizing optimal stochastic power management strategies. This formulation describes the *state* of the system, the *state constraints*, the available *controls* and the associated *control constraints* and the optimization *criterion*. After the formulation of the optimal control problem, we derive a *stochastic dynamic programming principle* from which one can obtain an algorithm for the synthesis of optimal strategies. This is the subject of Section 2.2. There, the optimal control problem is first stated into a deterministic framework (Section 2.3) to fix the main ideas and the stochastic formulation is introduced in Section 2.4.

The third stage concerns the simulation and interpretation of the performance of the obtained strategies. The appropriate simulation model and the performance of such controllers is discussed in section 2.5.

## 2.2 Stochastic Power Management

Stochastic power management stems from the idea of applying power management strategies to *real driving profiles* in a mathematically sound fashion. A central hypothesis the deterministic setting is that the vehicle will follow a known speed profile. This assumption is reasonable in some cases<sup>1</sup>. The considered application seeks to enrich the number of cases to which one can apply optimal power management strategies. Therefore, the general goal of optimal stochastic power management is to take into account uncertainties in the information about the vehicle state and/or route in the optimization routine.

---

<sup>1</sup>This modeling choice is relevant insofar that hybrid vehicles are expected to be equipped with embedded NAV systems that actually possess information about expected speeds and topological data. This information can be exploited in order to construct an expected speed profile with a certain degree of confidence. Moreover, the measurement of official emission and consumption levels of hybrid vehicles (indeed, any vehicle) is made using normalized driving cycles. In this measurement setting, a professional driver follows a recommended speed (and gear changes whenever applicable) as best as possible. In this case, deterministic models are deemed representative of the driving situation.

The considerations surrounding such uncertainties are an important matter. In the literature [48, 56, 52, 61] one can find three main assumptions concerning the information available to synthesize the controller:

1. All information is available at departure.

This leads to the formulation of deterministic strategies, as already discussed. The optimal controller can be synthesized in an open loop fashion, depending on the vehicle state and the position of the vehicle along the route.

2. Route-specific information is available at departure but the state of the vehicle in the future is uncertain.

In this case, since the route is known, the controller depends on the vehicle position and has a state feedback to adjust for the uncertainties occurring during the drive.

3. The route is unknown, as well as the vehicle state.

With this assumption, the controller obtained is independent of the vehicle position and has a feedback on the vehicle state.

The stochastic power management in this work considers the second level of information, namely, route-specific information with uncertain vehicle state. The random variable considered is the *vehicle future speed*. The vehicle speed enters in the optimization problem as a discrete-time Markov chain over the nodes on the route.

As mentioned, we use route-specific information available in the NAV in order to synthesize more robust power management strategies. To devise efficient strategies, the control synthesizer needs information about future power demands that are as close as possible to the actual power demand. Otherwise said, given a fixed route, the power management must be robust enough to take into account most of the situations occurring in a realistic driving cycle. In the case presented here, the strategy uses information from the NAV to better account for the uncertainty in the vehicle speed profile.

This approach is justified by the following rationale: the NAV speed profile represents to some extent the general lines of the driver behavior. It is judged realistic to consider that typically the driver will not exactly follow the suggested speed but will drive following a certain pattern depending on the NAV speed, as shown in Figure 2.1. As a result of the disparity between the actual driving speed - not known *a priori* - and the given NAV speed profile, the knowledge of how much power is required by the vehicle is not known with certainty. Each link's speed depends on particular characteristics of the route segment considered, such as type, number of lanes, location and also dynamic factors such as weather conditions or the hour of the day. These characteristics are relevant to the driver and judged to influence the “real” speed profile of the vehicle. Therefore, the NAV speed of each link should be taken into account in determining stochastic properties of future vehicle speeds.

The stochastic optimal control problem is stated in the discrete-time framework. The speed profile is modeled as a *stochastic process*. Let  $(\Omega, \mathcal{A})$  and  $(\Xi, \mathcal{F})$  be two measurable spaces. Define a random variable  $\xi : \Omega \rightarrow \Xi$ . We shall refer to  $\xi$  as the random variable and its realizations, the utilization of which will be clear from the context. This random variable models the speed at the next decision stage, i.e., the next link's entry node. Now, let  $\mathbb{P}$  be a probability over  $(\Xi, \mathcal{F})$ , so as to construct the probability space  $(\Xi, \mathcal{F}, \mathbb{P})$  and the process  $\{\xi\} = (\Xi, \mathcal{F}, (\xi_t)_{t \in \mathbb{N}}, \mathbb{P})$  with associated filtration<sup>2</sup>  $(\mathcal{F}_t)_{t \geq 0}$ .

---

<sup>2</sup>Quickly said, the filtration at node  $t$  can be seen as all the information available at time  $t$  when looking back to the process up to  $t$ . For the details on the construction of probability spaces, see [13].

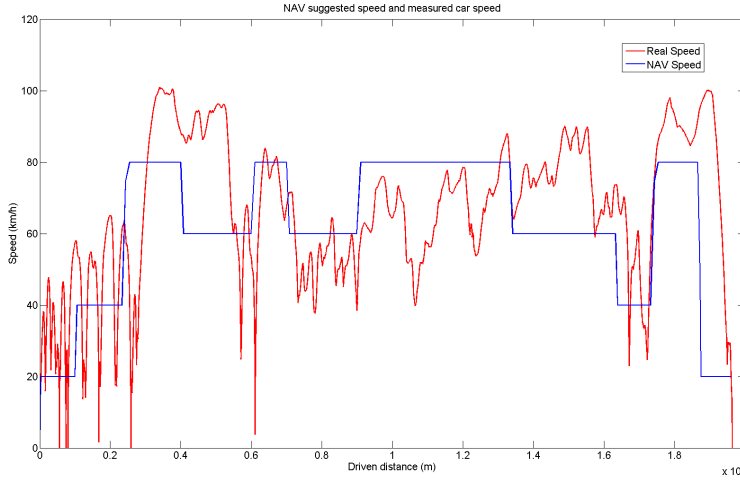


Figure 2.1 – Illustration of how the NAV speed and measured speed differ.

Let  $\xi_t$  be the known vehicle speed at node  $t$  and  $\Omega^{\text{NAV}}$  the NAV speed profile. The conditional probability of the term  $\xi_{t+1}$  of the process when having information about the process up to the  $k^{\text{th}}$  term is written as

$$\mathbb{P}(\xi_{t+1} \mid \mathcal{F}_t, \Omega^{\text{NAV}}). \quad (2.1)$$

From the Markov property, we have

$$\mathbb{P}(\xi_{t+1} \mid \mathcal{F}_t, \Omega^{\text{NAV}}) = \mathbb{P}(\xi_{t+1} \mid \xi_t, \Omega^{\text{NAV}}). \quad (2.2)$$

Intuitively, probability  $\mathbb{P}$  should depend on characteristics of the route, such as number of lanes, location and limiting speed. As mentioned, the NAV provides such information implicitly through the NAV speed profile. Therefore, one can include this information<sup>3</sup> in the conditional probability in order to “better” assess probabilities.

**Remark 2.1.** *The process  $\{\xi\}$  models the vehicle speed between nodes of the route. In this sense, the probability  $\mathbb{P}$  is the probability of realization of the vehicle speed when the vehicle passes the next node  $k + 1$ . In this model, the dynamic of the speed between nodes is, at first, neglected. In reality, speed  $\xi_t$  is viewed as the average speed between node  $t - 1$  and  $t$ . Therefore, the probabilities  $\mathbb{P}$  represent the likelihood of the average vehicle speed between node  $t$  and  $t + 1$ . A precise description of the probability model is discussed in Chapter 7.*

The immediate consequence of a modeling the vehicle speed as a stochastic process is that the vehicle power demand  $P_{\text{network}}$  is a random process. Indeed, the instant power imposed by the driver on the high-voltage network depends on the vehicle instant speed, acceleration and the terrain slope. In this study, the contribution of the terrain slope is subtracted and thus, the power demand depends only on the instant vehicle acceleration and speed.

**Remark 2.2.** *Removing the contribution of the inclination of the terrain is a modeling choice. The main difficulty of including terrain information in the algorithms is the unavailability of data. Indeed, the NAV system has information concerning topology of the routes, but the information is scarce and difficult to treat numerically. Scarcity means that terrain inclinations are not available for all links in the map database. It is hard to treat because the entry and exit nodes of a link are arbitrarily defined in the map database. Therefore, even if the inclination is available, it is not an immediate task to reconstruct the terrain topology for any given route. Since one can evaluate the energetic contribution for terrain inclination fairly easily, including this contribution directly in our analysis would result in a redundant complication and is, therefore, not taken into account.*

In this work, the acceleration model considered is a constant acceleration from speed  $\xi_t$  to speed  $\xi_{t+1}$ . More evolved acceleration models can be constructed to take in account the energetic

<sup>3</sup>This is treated as exogenous information because it does not belong to the filtration of process  $\{\xi\}$ .

contribution of intra-link speed dynamics, but this is judged redundant based on our modeling choices (cf. Remark 2.1).

## 2.3 Deterministic Dynamic Programming

A first step towards the synthesis of more sophisticated power management strategies consists in analyzing deterministic feedback strategies. Deterministic strategies are obtained if route-specific information is available and the vehicle state evolution is known with certainty. In the case of the model considered in this work, a deterministic strategy assumes that the speed profile, and thus the power profile, is known with certainty.

A discrete-time setting is used in the formulation of the optimal control problem in this chapter.

We recall the main points of the model, described in detail in Chapter 1 with a slightly different notation. The nodes are indexed by  $t = 1, \dots, T$ . The state is the battery SOC  $x_t \in \mathcal{X} := [0, 1]$ , the range extender state (on or off)  $q_t \in \mathcal{Q} := \{0, 1\}$  and the time since the last switch (the switch lock variable)  $\tau_t \in \mathcal{T} := [0, T']$ , where  $T' > 0$  is a time greater than the time of the trip duration.

**Remark 2.3.** *The objective of this section is to use a dynamic programming (DP) algorithm to synthesize the optimal control sequence. In this section, we consider that the RE has a unlimited amount of fuel. The consequence is that the mass of fuel in the tank is not a state variable. This modeling choice is justified by the following rationale: as mentioned in Chapter 1, the objective adding a RE module to an EV is to ensure that the driver reaches its target. At the same time, the objective of the optimization is to reduce the cost of the trip. In current prices, this amounts in reducing the usage of the RE. In this sense the interesting scenarios to use a stochastic strategy are those in which the vehicle cannot reach the target in EV mode and must use the RE. This situation can be achieved by setting the initial SOC to a convenient value with respect to the total drive distance<sup>4</sup>. Even in the limiting cases in which the fuel tank must be completely depleted to ensure arriving at the destination point, we can a posteriori rule out strategies that consume more than the available fuel. This simplification in the model allows for a substantial reduction in computational time in synthesizing optimal controllers, because of the curse of dimensionality in DP algorithms.*

In order to shorten the notation, the state components are grouped in a *hybrid state vector* denoted by  $z_t := (x_t, \tau_t, q_t) \in \mathcal{Z} := \mathcal{X} \times \mathcal{T} \times \mathcal{Q}$ . The vehicle speed  $y_t \in (0, \bar{y}]$  and the link length  $d_t \geq 0$  are known with certainty. They are treated as exogenous information, e.g., given by a navigation system or imposed by a driving cycle.

The continuous control consists of the range extender output power  $u_t \in \mathcal{U}_t \subset [\underline{u}, \bar{u}]$  where  $\underline{u}, \bar{u} > 0$  are the continuous control lower and upper bounds respectively. We remark that the continuous control space  $\mathcal{U}$  depends on the vehicle speed  $y_t$ . But since  $y_t$  is known with certainty, this dependence is seen through the node index  $t$ . The switch control is the switch decision  $w_t \in \mathcal{W} := \{0, 1\}$ .

**Remark 2.4.** *The switch control introduced above has a different structure from the one introduced in (1.55). Now, we consider that at a node  $t$ , a discrete order  $w_t = 1$  consists of a switch, i.e. a change in the discrete state, whereas  $w_t = 0$  is a no switch, i.e. no modification of the discrete state.*

<sup>4</sup>Very roughly, 1% of SOC corresponds to 1km. In simulation, if one consider a, say, 20km driving cycle, the initial SOC is set at 20% in order to achieve a situation in which a prudent control strategy should use the RE at some point. If the driving cycle considered has 50km, the departing SOC is set at 50%, accordingly, preserving the scenario of interest.

The discrete-time dynamics of the hybrid state components is given by:

$$x_{t+1} = f_t(x_t, q_t, u_t), \quad (2.3a)$$

$$\tau_{t+1} = (1 - w_t)\tau_t + 2\frac{d_t}{y_t + y_{t+1}}, \quad (2.3b)$$

$$q_{t+1} = (1 - w_t)q_t + (1 - q_t)w_t. \quad (2.3c)$$

Equation (2.3a) represents the non-linear decrease in the SOC as a function of the SOC itself and range extender state. A particular battery model is used to determine the specific rate of SOC change (in %/s) while the node dependence reflects the total time spent on the link, affected by the vehicle speed and link's length. The switching effect of control  $w_t$  on the state dynamics is represented in equation (2.3c). A decision of no switch  $w_t = 0$  maintains the value of the discrete state while a switch  $w_t = 1$  moves the discrete state to its complement  $1 - q_t$ . The switch control also intervenes on the switch lock through relation (2.3b) by resetting the value of  $\tau$  whenever a switch is decided. The decision lag constraint, that imposes a time interval of at least  $\delta > 0$  between consecutive switches, can be stated in terms of the trajectory of the state variable  $\tau_t$ .

The *hybrid control* is denoted by  $a_t := (u_t, w_t)$  with values in  $\mathcal{A}_t := \mathcal{U}_t \times \mathcal{W}$ . This notation enables us to abbreviate the set of equations (2.3) conveniently as

$$z_{t+1} = h_t(z_t, a_t).$$

Although hybrid controls are defined in a set of admissible values, the control sequence  $(a_1, \dots, a_T)$  might not respect the decision lag constraint of switching decisions. The next definition introduces the set of *admissible hybrid control sequence* that respect the lag constraint.

**Definition 2.1.** *The control sequence  $a := (a_1, \dots, a_T)$  is an admissible hybrid control sequence if it satisfies the relations*

1. For  $(z_1, \dots, z_T) \in \mathcal{Z}^T$ , we have  $a \in \mathcal{A}_1 \times \dots \times \mathcal{A}_T$ .
2. Let  $S := \{t \in \{1, \dots, T-1\} \mid w_t = 1\}$  be the set of links in which a switch decision is made. Then,  $\tau_t \geq \delta$  for  $t \in S$ .

Given  $T > 0$ , denote by  $\mathcal{A}^T$  the set of all admissible hybrid controls containing  $T$  decisions, i.e.,  $\mathcal{A}^T := \{a = (a_1, \dots, a_T) \mid a \text{ verifies 1. and 2.}\}$ .

The first condition states that every hybrid control should be in the hybrid control domain. The second condition is the decision lag imposed on the value of variable  $\tau_t$  at nodes in which a switch decision is made.

At each node, a hybrid decision incurs a fuel cost. The fuel cost can be separated into a running cost  $l_t(u_t)$  and a switching cost  $\rho(q_t, w_t)$ . The running cost represents the amount of fuel consumed in the range extender to output power  $u_t$ . The term in function  $l_t$  contains a specific fuel consumption (typically in g/s) that depends on the type of range extender considered. Also, the running cost function depends on the time spend on the link and thus, on  $y_t, t_{t+1}$  and  $d_t$ . This dependence is included in the non-autonomous character of  $l_t$ . The switching cost  $\rho$  represents a constant start-up cost of the range extender. There is no turn off cost.

As discussed above, the optimization criteria considered in this section is the total *economic* cost of operation of the range extender during driving cycle. To construct such a criteria, the cost of electric energy must be included in the criterion. Since the battery SOC – which can be seen as the image of the energy available on the battery – is a state variable, the total electric energy consumption can be obtained through the SOC final value,  $x_{T+1}$ . Then, given an initial SOC  $x_1$ , the total electric energy consumption, in % of the maximum energy capacity, is given by  $x_1 - x_{T+1}$ .



In the deterministic setting, given  $T > 0$ , initial conditions  $z_1 = z \in \mathcal{Z}$  and an admissible hybrid control sequence  $a \in \mathcal{A}^T$ , the set of equations (2.3) can be used to determine the trajectory of the system with certainty. The total (economic) cost of operation, that we wish to minimize, for the hybrid control sequence  $a \in \mathcal{A}^T$ , starting at node  $t = 1$  with hybrid state  $z_1 := z = (x_1, \tau_1, q_1)$  is defined as

$$J_1(z, a) = \sum_{t=1}^T l_t(u_t) + \rho(q_t, w_t) + \beta(x_1 - x_{T+1}). \quad (2.4)$$

As discussed before,  $\beta > 0$  is a parameter that represents the price ration of electric energy and fuel. Such a conversion applied to current price estimates yields  $\beta = 0.44 \text{ €/MJ}$ .

Now we proceed to state the optimal control problem associated to the problem of finding a hybrid control sequence that minimizes the total operating cost of the vehicle. Set  $T > 0$ . For an arbitrary starting node  $t \in \{1, \dots, T\}$ , giving initial conditions  $z_t := z = (x_t, \tau_t, q_t)$  and an admissible hybrid control sequence  $a \in \mathcal{A}^{T-t}$  the total operating cost is

$$J_t(z, a) = \sum_{k=t}^T l_k(u_k) + \rho(q_k, w_k) + \beta(x_t - x_{T+1}). \quad (2.5)$$

The value function is then defined as the cost of the optimum control sequence:

$$v_t(z) = \inf_{a \in \mathcal{A}^{T-t}} J_t(z, a). \quad (2.6)$$

The next proposition states that the value function satisfies a dynamic programming principle.

**Proposition 2.1.** *The value function defined in (2.6) verifies, for all  $z \in \mathcal{Z}$ ,  $t = T, \dots, 1$ ,*

$$v_{T+1}(z) = 0, \quad (2.7a)$$

$$v_t(z) = \inf_{a \in \mathcal{A}_t} l_t(u) + \rho(q, w) + \beta(x - x_{t+1}(x, a)) + v_{t+1}(h_t(z, a)). \quad (2.7b)$$

*Proof.* The proof is derived from simple recursion (see [15] for details).  $\square$

In the deterministic setting, the number of state variables – the dimension of the state – is 3. A state of dimension 3 is deemed acceptable for equations (2.7) to be solved via a dynamic programming algorithm. This algorithm constructs a grid by discretization of the state space. The dynamic programming algorithm can be broadly divided in two parts: the computation of the value function and the computation of an optimal hybrid control sequence. One can compute the value function using equations (2.7), which are solved backwards for all values of  $z$  in the grid, coupled with some interpolation method whenever a value of  $z_{t+1}$  falls outside of the grid. Then, given an initial decision at the first node, the hybrid control is sequentially chosen such that it minimizes the total operating cost  $l_t(u) + \rho(q, w) + \beta(x - x_{t+1}(x, a)) + v_{t+1}(h_t(z, a))$ . This procedure is detailed in the algorithm that follows.

**Algorithm 2.1** (Dynamic Programming Algorithm). *Step 1: Compute value function.*

1.1 Set  $t = T + 1$ , set  $\beta$  and construct a discretization of the state space  $\mathcal{Z}^\#$ ;

1.2 For all  $z \in \mathcal{Z}^\#$  set  $v_t(z) = 0$ ;

1.3 For  $t = T, \dots, 1$ , for all  $z \in \mathcal{Z}^\#$  compute

$$v_t(z) = \min_{a \in \mathcal{A}_t} l_t(u) + \rho(q, w) + \beta(x - x_{t+1}(x, a)) + v_{t+1}(h_t(z, a)).$$

Once having evaluated  $v$  in  $\mathcal{Z}^\#$  at all links the optimal trajectory can be synthesized as follows:

*Step 2: Control synthesis.*

2.1 Set  $t = 1$ , and initial condition  $z_t = z$ ;

2.2 Compute hybrid control:

$$a_t^* = \arg \min_{a \in A_t} l_t(u) + \rho(q, w) + \beta(x - x_{t+1}(x, a)) + v_{t+1}^\#(h_t(z, a))$$

2.3 Update state to  $z \leftarrow h_t(z, a_t^*)$ , set  $t \leftarrow t + 1$ . If  $t = T + 1$  terminate the algorithm, otherwise go to 2.2.

Here,  $v^\#(h_t(z, a))$  denotes an interpolate of the value function  $v$  at point  $h_t(z, a)$ , which may fall out of the points in grid  $\mathcal{Z}^\#$ . The control sequence  $a^* = (a_1^*, \dots, a_T^*)$  obtained using algorithm 6.1 is only an approximation of the global optimum trajectory since one only considers the value function at discrete points of the state space. Nevertheless, it can be showed that the control sequence  $a^*$  converges to the optimum control sequence as the grid covering the state space becomes finer, covering all  $\mathcal{Z}$  in the limit. As so, for a grid fine enough, one cannot expect to outperform a control sequence  $a^*$  synthesized using algorithm 6.1.

**Remark 2.5.** We draw the attention to the fact that one needs to compute the value function for all points in  $\mathcal{Z}^\#$  for all nodes. Denoting by  $D$  the dimension of  $\mathcal{Z}^\#$ ,  $D := \dim(\mathcal{Z})$ , and supposing the grid is uniform in all directions with  $N_Z$  points, the total number of value function computations is  $T \times N_Z^D$ . The exponential character in the dimension of the state space for the computation of the value function is better known as the curse of dimensionality. This "curse" makes the application of the dynamic programming algorithm prohibitively expensive even for relatively low-dimension states of dimension 7 or 8. When such is the case, one can still recourse to parallelization techniques and/or machine power to compute the value function on  $\mathcal{Z}^\#$ . Nevertheless, for applications with state dimension of 3 or 4, such as ours, the dynamic programming algorithm can be expected to have a reasonable performance in commercial desktop PCs.

Before concluding this section, one must stress that the performance levels of controls synthesized using a dynamic programming algorithm in the deterministic setting are only valid if the evolution of the hybrid state vector is dictated by function  $h_t$ . More precisely, the supposedly known speed profile  $y_t$  engenders a vector field  $h_t$  that then is used in the backward computation of the value function at points of  $\mathcal{Z}^\#$ . Then, once the value function is computed over  $\mathcal{Z}^\#$ , given one initial state  $z$ , one can perform an *open-loop* synthesis of an approximation optimal controller. However, in real driving conditions, it is highly unlikely that the state vector trajectory will follow the prescribed dynamics  $h_t$ . In this case, the value function computed in step 1 does not correspond in any way to the optimal cost and there is no guarantee whatsoever that the control sequence will have a performance as expected. A possible way to counter some of these degrading effects on the controller is to close the loop on the state, by measuring it at each node  $t$ . Even proceeding as such, the value function still does not represent the optimal cost, the system only is refreshed with new information about the state at each node, an improvement with respect to the open-loop case. This closed-loop solution might reveal itself interesting from the point of view of a practical implementation, but has no guarantee of optimality in any sense. Due to the practical interest they represent, simulation of the performance levels of a deterministic closed-loop controller are performed and the results are presented in section 2.5 below.

## 2.4 Stochastic Dynamic Programming

After presenting the set-up of a deterministic approach for the synthesis of power management synthesis in section 2.3, we focus on the synthesis of *stochastic power management strategies*. We start by recalling the motivation for such an approach. Then, the stochastic optimal control problem is presented along with a dynamic programming principle verified by the associated value function. Finally, we give a description of the algorithm used to synthesize feedback strategies, namely, the *stochastic dynamic programming algorithm*.

As highlighted in the preceding section, deterministic power management strategies suppose that the vehicle power demand profile is known with certainty. This amounts to suppose that the speed profile of the vehicle is known with certainty, since we consider a flat terrain. Although this assumption may result in a tolerable approximation in certain controlled cases, one cannot expect to have an exact knowledge of the vehicle speed profile in general situations. Typically, in a routine usage of a hybrid vehicle, the speed developed by the driver cannot be predicted with enough precision to assume that the speed is known with certainty. An immediate consequence is that in a routine usage of a hybrid vehicle the emission levels and/or fuel costs tend to be higher than the official declared levels if the official levels are obtained using global optimization techniques. This consideration motivates the development of power management system that are more "robust" with respect to the driving conditions encountered by a hybrid vehicle user. Modeling the evolution of the vehicle speed as a random process allows the evolution of the hybrid state to be treated as random process. In this case, the notion of optimality of a controller differs conceptually from the deterministic notion of optimality.

In the stochastic setting, the state variables are the battery SOC  $x_t \in \mathcal{X} := [0, 1]$ , the range extender on/off state  $q_t \in \mathcal{Q} = \{0, 1\}$ , the switch lock  $\tau_t \in \mathcal{T} := [0, \delta]$  and the vehicle instant speed  $y_t \in \mathcal{Y} := (0, \bar{y}]$ . Similarly, for convenience, we denote the hybrid state vector by  $z_t := (x_t, y_t, \tau_t, q_t) \in \mathcal{Z} := \mathcal{X} \times \mathcal{Y} \times \mathcal{T} \times \mathcal{Q}$ . With respect to the notation introduced earlier, since  $\xi_t$  is  $\mathcal{F}_t$  measurable, we set at node  $t$ ,  $y_t \equiv \xi_t$ . This is only done to homogenize the notation with that introduced in the last section.

Hybrid controls comprise the continuous control  $u_t \in \mathcal{U}(y_t) \subset [\underline{u}, \bar{u}]$  representing the range extender output power and the discrete switch control  $w_t \in \mathcal{W} := \{0, 1\}$  applied to the system at nodes  $t = 1, \dots, T$ . (Notice that the dependency of the control domain on the vehicle speed, via (1.54), is made explicit). The hybrid control is denoted by  $a_t = (u_t, w_t)$  valued in  $\mathcal{A}(z_t) = \mathcal{U}(y_t) \times \mathcal{W}$ .

In the stochastic setting, a working hypothesis is that the hybrid control sequence satisfies a *non-anticipative constraint*:

$$a_t \preceq \mathcal{F}_t. \quad (2.8)$$

This constraint imposes decision  $a_t$  to be  $\mathcal{F}_t$  measurable (for a more detailed account, we refer to [78]). The interpretation is that the decision  $a_t$  is a function only of the past values of the process  $\{\xi\}$  up to  $t$ , and not of future realizations. After the decision  $a_t$  is made, a realization of the random variable  $\xi_t$  becomes know, the decision cost is incurred and the process is repeated. This decision process is summarized in the causality diagram below:

$$\text{decision } a_t \rightsquigarrow \text{realization } \xi_{t+1} \rightsquigarrow \text{decision } a_{t+1} \rightsquigarrow \dots$$

Given a state at node  $t$  and a hybrid control  $a_t = (u_t, w_t)$ , the hybrid state vector at node  $t + 1$  is a random variable given by the dynamics:

$$X_{t+1} = f_t(x_t, q_t, u_t, \xi_{t+1}), \quad (2.9a)$$

$$\Upsilon_{t+1} = (1 - w_t)\tau_t + 2\frac{d_t}{y_t + \xi_{t+1}}, \quad (2.9b)$$

$$q_{t+1} = (1 - w_t)q_t + (1 - q_t)w_t. \quad (2.9c)$$

These dynamics are abbreviated using the notation  $Z_{t+1} = h(z_t, a_t, \xi_{t+1})$ .

Given the nonanticipativity constraint (2.8) at the beginning of the process, one cannot build a sequence of admissible controls depending on future values of the state. A control may however plan in advance what action it should take if a given random realization occurs. This leads to the notion of a *hybrid control policy* (or *hybrid control strategy*) defined next.

**Definition 2.2.** *Given a hybrid state  $z \in \mathcal{Z}$ , define an admissible control law  $\alpha$  as a mapping of the state space in the admissible hybrid control set:*

$$\begin{aligned} \alpha &: \mathcal{Z} \rightarrow \mathcal{A} \\ z &\mapsto \alpha(z) = a \end{aligned} \quad (2.10)$$

Given  $T > 0$ , an admissible hybrid control policy  $\pi$  is defined as a sequence of admissible control laws  $\pi := (\alpha_1, \dots, \alpha_T)$ . We denote by  $\Pi_t$  the set of all admissible hybrid control policies containing  $t$  control laws.

Now, given an initial hybrid state  $z$  and a hybrid policy  $\pi$ , the hybrid state process starting at  $z$  and controlled by policy  $\pi$  is given by

$$Z_1 = z, \quad Z_{t+1} = h_t(Z_t, \alpha_t(Z_t), \xi_{t+1}), \quad t = 1, \dots, T \quad (2.11)$$

and the total expected cost of the process solution of (2.11) is

$$J_1(z, \pi) = \mathbb{E} \left[ \sum_{t=1}^T l_t(u_t) + \rho(q_t, w_t) + \beta(x_1 - X_{T+1}) \right]. \quad (2.12)$$

The power management strategy in the stochastic setting is a hybrid control policy that minimizes the *expected total cost* of operation. More precisely, policy  $\pi^*$  is said to be optimal if, given a initial state, the policy achieves

$$\pi^* \in \arg \inf_{\pi \in \Pi_T} J_1(z, \pi).$$

For an arbitrary starting point  $t \in \{1, \dots, T+1\}$ , initial conditions  $z = (x, y, \tau, q)$  and a control policy  $\pi$ , the total expected operating cost of process  $\{Z\}$  is

$$J_t(z, \pi) = \mathbb{E} \left[ \sum_{k=t}^T l_k(u_k) + \rho(q_k, w_k) + \beta(x_t - X_{T+1}) \mid \mathcal{F}_t, \Omega^{\text{NAV}} \right]. \quad (2.13)$$

We define the *value function* associated with the minimization of the cost (2.13) as

$$v_t(z) = \inf_{\pi \in \Pi_{t-T}} J_t(z, \pi). \quad (2.14)$$

To obtain a dynamic programming principle, we start by conditioning the criterion to the filtration  $\mathcal{F}_T$ , minimizing with respect to the control policy; obtaining the following structure

$$\begin{aligned} \inf_{a_1} \mathbb{E} & \left[ l_1(u_1) + \rho(q_1, w_1) + \beta(x_1 - X_2) + \inf_{a_2} \mathbb{E} \left[ l_2(u_2) + \rho(q_2, w_2) + \beta(X_2 - X_3) \right. \right. \\ & \left. \left. + \dots + \inf_{a_T} \mathbb{E} \left[ l_T(u_T) + \rho(q_T, w_T) + \beta(X_T - X_{T+1}) \mid \mathcal{F}_T \right] \dots \mid \mathcal{F}_2 \right] \mid \mathcal{F}_1 \right], \end{aligned} \quad (2.15)$$

where we omit the conditioning with respect to  $\Omega^{\text{NAV}}$ .

This imbricated structure enables the computation of the value function in a sequential backward manner. The next proposition details the dynamic programming principle for the value function defined in (2.14).

**Proposition 2.2.** *The value function defined in (2.14) verifies, for all  $z \in \mathcal{Z}$ ,  $t = T, \dots, 1$ ,*

$$v_{T+1}(z) = 0, \quad (2.16a)$$

$$v_t(z) = \inf_{a \in \mathcal{A}(z)} l_t(u) + \rho(q, w) + \mathbb{E}[\beta(x - X_{t+1}) + v_{t+1}(h_t(z, a, \xi)) \mid \mathcal{F}_t]. \quad (2.16b)$$

Proposition 2.2 is the stochastic counterpart to proposition 2.1. In equation (2.16b) one can readily see that the control choice at stage  $t$  has the objective of optimizing the *expected cost* of the process from time  $t + 1$  to  $T$ . If for any reason process  $\{Z\}$  can be known with certainty, then one can simply drop the expected value in (2.16b) and replace the realization of  $\xi$  by its certain value. The resulting deterministic dynamic programming principle in proposition 2.1 is then recovered.

The robust character of a stochastic optimal policy, with respect to a deterministic one, stems from the fact the one is optimizing the average of the process. In a sense, each control decision  $a_t$  "takes into account" all possible realizations of  $\xi_t$ , weighted by the probability of each realization, to steer the system in a cost-optimal fashion. Such an optimization procedure can be justified by the fact that the process  $\{Z\}$  is often repeated and a the optimal policy  $\pi^*$  can be applied several times. For one particular realization of the process  $\{Z\}$  starting at  $z$ , the total operation cost *a posteriori* obtained by a stochastic optimal policy  $\pi^*$  may be greater than or lower than the expected optimal cost  $v_1(z)$ . However, after a great number of realizations of  $\{Z\}$ , each controlled by an stochastic optimal policy  $\pi^*$ , the total operating cost *average* is close to the optimal value  $v_1(z)$ .

The value function can be calculated using the stochastic dynamic programming principle stated in proposition 2.2.

The stochastic dynamic programming algorithm is very similar to the deterministic dynamic programming algorithm, the main difference being in the evaluation of the expected value of the future cost. The first step is to construct a discrete grid  $\mathcal{Z}^\#$  of the state space, the idea being to compute the value function at the points of this grid. A discretization of the set  $\Xi$  is also necessary. We detail this point further.

We recall that the vehicle speed in the future is modeled as a random variable  $\xi : \Omega \rightarrow \Xi$ , where  $(\Omega, \mathcal{A})$  is a measurable space and  $(\Xi, \mathcal{F}, \mathbb{P})$  is a probability space. So far, no assumptions were made about the properties of  $\Xi$ . For instance,  $\Xi$  can be a discrete or continuous set, finite or infinite. The characteristics of  $\Xi$  depends on the model used for representing the random variable  $\xi$ . However, at some point, a discretization of  $\Xi$  is needed to deal numerically with the random variable  $\xi$ . This discretization can be done before or after the construction of a model of  $\xi$ . We denote by  $\Xi^\#$  the discretization of  $\Xi$ , the image space of the random variable  $\xi$ . We denote the number of elements of  $\Xi^\#$  as  $N_\Xi$ . Once  $\Xi^\#$  is constructed, the expected value operator can be replaced by its discrete counterpart. In our application  $\Xi^\#$  is constructed from a discrete set, i.e.  $\Xi$  is already discrete and  $\Xi^\# \subset \Xi$ . In this case the discrete expected value of a (measurable) function  $h$  of the random variable  $\xi$  is given by

$$\mathbb{E}[h(\xi)] = \sum_{i=1}^{N_\Xi} h(\xi_i) \mathbb{P}(\{\xi = \xi_i\}). \quad (2.17)$$

We point to [91, 7, 6] for details on quantization methods and error analysis.

Having a discretization of the state space, a discretization of  $\Xi$  and a way of computing the expect value, the stochastic dynamic programming algorithm computes the value function for all points in  $\mathcal{Z}^\#$  in backward fashion and then synthesizes the optimal feedback. This procedure is detailed next.

**Algorithm 2.2** (Stochastic Dynamic Programming Algorithm). *Step 1: Compute value function.*

1.1 Set  $t = T + 1$ , set  $\beta$  and construct  $\mathcal{Z}^\#, \Xi^\#$ ;

1.2 For all  $z \in \mathcal{Z}^\#$  set  $v_t(z) = 0$ ;

1.3 For  $t = T, \dots, 1$ , for all  $z \in \mathcal{Z}^\#$  compute

$$v_t(z) = \min_{a \in \mathcal{A}(z)} l_t(u) + \rho(q, w) + \sum_{\xi \in \Xi^\#} \left( \beta(x - f_t(x, q, u, \xi)) + v_{t+1}(h_t(z, a, \xi)) \right) \mathbb{P}(\xi).$$

Once having evaluated  $v$  in  $\mathcal{Z}^\#$  at all links the optimal trajectory can be synthesized as follows:

*Step 2: Feedback control synthesis.*

2.1 Set  $t = 1$ , and initial condition  $z_t = z$ ;

2.2 Compute hybrid control:

$$a_t^* = \arg \min_{a \in \mathcal{A}(z)} l_t(u) + \rho(q, w) + \sum_{\xi \in \Xi^\#} \left( \beta(x - f_t(x, q, u, \xi)) + v_{t+1}^\#(h_t(z, a, \xi)) \right) \mathbb{P}(\xi);$$

2.3 Draw a realization  $\xi^* \in \Xi$ ;

2.4 Update state to  $z \leftarrow h_t(z, a_t^*, \xi^*)$ , set  $t \leftarrow t + 1$ . If  $t = T + 1$  terminate the algorithm, otherwise go to 2.2.

Step 2.3 can be seen as a feedback coming from the environment. As the process  $\{\xi\}$  evolves, more information can be add to the synthesis of future controls. In the case of the synthesis of a power management strategy, the random variable  $\xi_t$  models the (random) vehicle speed at node  $t + 1$ . Once the vehicle arrives at node  $t + 1$  the speed can be measured and this information can be added to the synthesis of the next control. The measured speed corresponds to a particular realization of the random variable  $\xi_t$  and counts as information that can be used by the system. In the algorithm, this information is used to compute the state vector at node  $t + 1$  in step 2.3.

Now we turn our attention to the existence of an optimal trajectory. We introduce the concept of feasible state:

**Definition 2.3.** Given  $t > 0$ , a state  $z \in \mathcal{Z}$  is said to be feasible for the stage  $t$  if there exists at least one admissible control input  $a$  such that  $h_t(z, a, \xi) \in \mathcal{Z}$  for all realizations of  $\xi$ . More precisely, given a decision stage  $t$ , the set of all feasible states is defined as

$$\text{feas}(t) = \{z \in \mathcal{Z} \mid \exists a \in \mathcal{A}(z), h_t(z, a, \xi) \in \mathcal{Z}, \forall \xi \in \Xi\} \quad (2.18)$$

The optimization problem has state constraints and the state must remain inside an admissible region  $\mathcal{Z}$ . A standard approach to deal with the state constraints in the dynamic programming algorithm is to penalize the control every time the state falls outside  $\mathcal{Z}^\#$  by setting  $v_t(z) = \infty$ , if  $z \notin \mathcal{Z}$  (or a numerical equivalent). (We recall that one must always be cautious when handling the "infinity" in numerical simulations, in particular when interpolating the value function near the border of  $\mathcal{Z}$ ). In this direction, the grid  $\mathcal{Z}^\#$  must be slightly larger than  $\mathcal{Z}$  and the value function is initialized as  $v_T(z) = 0$  if  $z \in \text{feas}(T) = \mathcal{Z}$  and  $v_T(z) = \infty$  otherwise.

When proceeding backwards to stage  $T - 1$ , given a state  $z$ , feasibility demands that exists at least one admissible control  $a$  such that  $h_{T-1}(z, a, \xi) \in \mathcal{Z}$  for all possible realizations of  $\xi$ . One can readily see that the feasibility property is preserved when the method of penalizing the value function is applied. Indeed, given a state  $z \in \mathcal{Z}$ , if for all admissible controls  $a$  is there is at least one realization

$\xi^-$  such that  $h_{T-1}(z, a, \xi^-) \notin \mathcal{Z}$ , the expectation is set to  $\infty$ . Then, the value function at this point is also penalized,  $v_{T-1}(z) = \infty$  and the non-feasibility information (points of  $\mathcal{Z}$  at which the value function is  $\infty$ ) is correctly propagated to subsequent stages.

In terms of the vehicle application, the existence of an optimal (indeed any) trajectory is related to the *maximum autonomy* of the vehicle. The case of having an empty feasible set at the destination point indicates that irrespectively of the control applied, the vehicle cannot reach its destination. A study about the maximum range of hybrid vehicles is developed in Chapter 5.

## 2.5 Simulation Results

This section is dedicated to the results obtained from simulations of synthesized optimal deterministic and stochastic controllers. The section is divided into two parts. The first deals exclusively with the deterministic algorithm and the second presents results for the stochastic algorithm.

### 2.5.1 Deterministic Controllers

Our goal here is to discuss the synthesis of deterministic controllers. We recall that in the deterministic case, the vehicle speed evolution is assumed to be known with certainty. The value function is then computed and a controller is synthesized using the dynamic programming Algorithm 6.1.

The first part of this subsection deals with alternative ways of treating the decision lag constraint imposed on the switching decision sequence. As pointed out, the main drawback of a dynamic programming algorithm is the curse of dimensionality, in which resource needs grows exponentially with the dimension of the state. In order to deal with the decision lag constraint, our model introduces the switch lock state variable  $\tau$ . Since the CPU time of a dynamic programming algorithm can be sensibly reduced by lowering the number of state variables, this work explores alternative ways of introducing the decision lag constraint without adding a state variable.

One important aspect is to avoid fast ICE switching. Since the intermittent fast switching is undesirable, a possible approach is to penalize the switch cost. In this direction, we introduce a penalty  $\lambda \geq 1$  in the switch cost, effectively considering the cost to be  $\lambda \rho(\cdot)$ . In Figure 2.2 we observe, the SOC trajectory for different values of  $\lambda$ .

The CPU time is of 95.5 seconds for all values of  $\lambda$ .

Table 2.1 shows the RE operation cost and number of switches obtained for several values of the penalization factor. For  $\lambda = 1$  the system is in its nominal cost configuration, but does not take into account the lag constraint in the discrete decision sequence. Penalizing the switch cost reduces indeed the number of switches of a strategy. Approximately, in the range  $\lambda \in [10, 50]$  the optimal strategy consists of 9 switches and for  $\lambda > 50$  the strategy seems to stabilize at 8 switches. The penalization approach has the practical advantage of simple numerical implementation and relatively fast execution time (95s). It also effectively reduces the number of total switches. The downside is that the technique does not see to that the control respects the lag condition. Even if there are fewer switches, they are not necessarily spaced enough in time to respect the decision lag.

As discussed before, in the deterministic setting, the optimal controller synthesis is done in an open-loop fashion. In other words, when using a specific speed profile to compute the value function the same speed profile must be used to the controller synthesis (step 2 of Algorithm 6.1). Therefore,

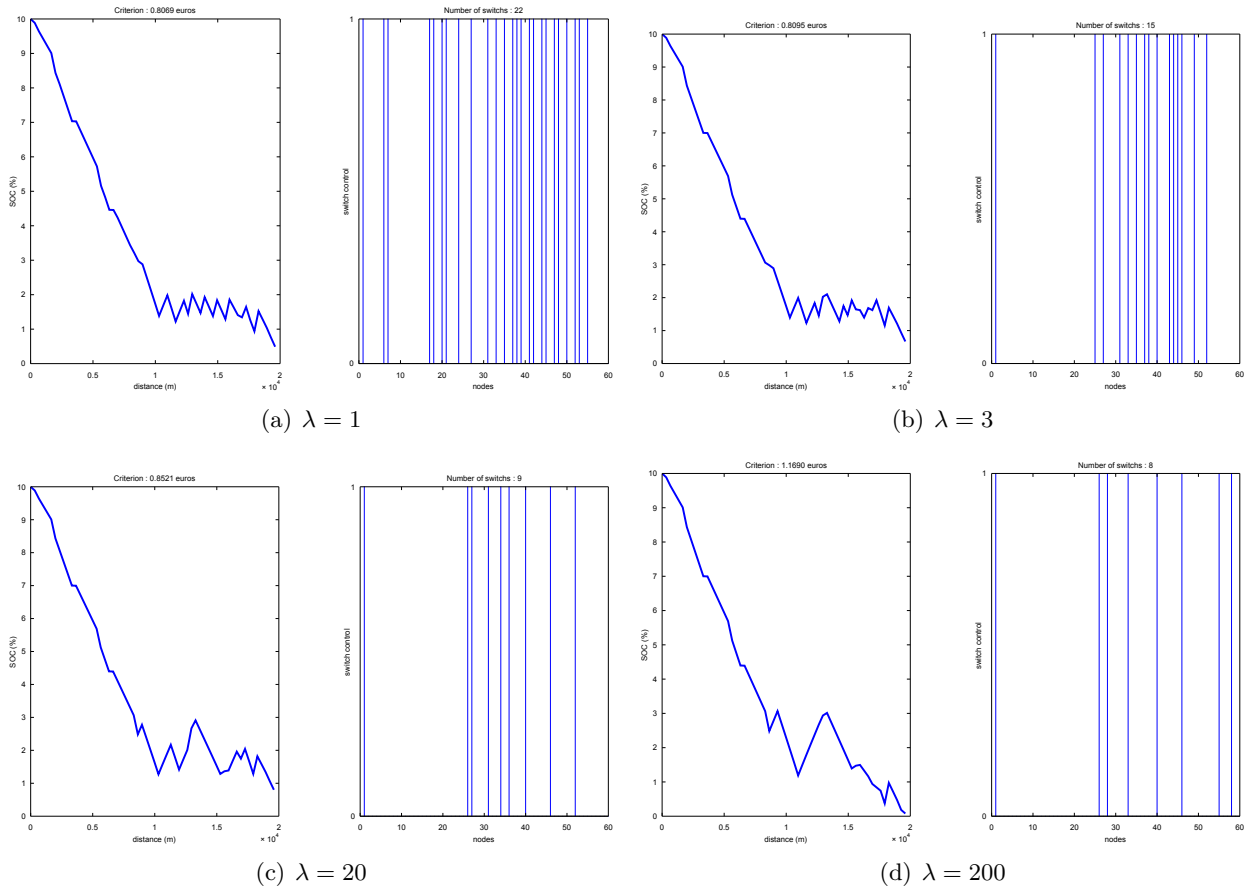


Figure 2.2 – State and control trajectories for deterministic controllers using a switch cost penalty factor  $\lambda$ .

Penalization	Operation cost (€)	switchs
$\lambda = 1$	0.8069	22
$\lambda = 3$	0.8095	15
$\lambda = 15$	0.8211	9
$\lambda = 20$	0.8521	9
$\lambda = 50$	0.8690	8
$\lambda = 100$	0.9690	8
$\lambda = 200$	1.1690	8

Table 2.1 – Performance criteria for deterministic controllers using a switch penalization factor  $\lambda \geq 1$ .



since the speed is known with certainty, the state can be integrated with certainty and no deviations are to be expected. This is what enables the open-loop synthesis, any feedback being unnecessary.

As mentioned, this approach is interesting in many cases such as normative cycles and bus lines, where the speed can be known with a relatively high degree of confidence. This remark motivates the next heuristic approach to make the controller more robust with respect to the minor deviations. The approach consists in feed-backing measured information to the controller, hoping the controller is able to “correct” the trajectory towards the supposed optimal one. Consider the implementation of the following closed-loop algorithm:

**Algorithm 2.3.** *Step 1: Compute value function as in Algorithm 6.1.*

*Step 2: Feedback control synthesis.*

2.1 Set  $t = 1$ , and initial condition  $z_t = z$ ;

2.2 Compute hybrid control:

$$a_t^* = \arg \min_{a \in \mathcal{A}_t} l_t(u) + \rho(q, w) + \beta(x - x_{t+1}(x, a)) + v_{t+1}^\#(h_t(z, a))$$

2.3 Feedback next state value  $z_1$  and update state to  $z \leftarrow z_1$ , set  $t \leftarrow t + 1$ . If  $t = T + 1$  terminate the algorithm, otherwise go to 2.2.

Observe that the crucial difference between Algorithms 6.1 and 2.3 is that in step 2.3, the state is updated with information coming from the feedback  $z_1$ , not from the model  $h$ . In both cases, the control decision is obtained by minimizing the cost at the next node using the model  $h_t$  (step 2.2), but in the close-loop version, the *de facto* next state comes from the feedback, not from the model  $h_t$ . This approach is compared against other feedback strategies in the next subsection.

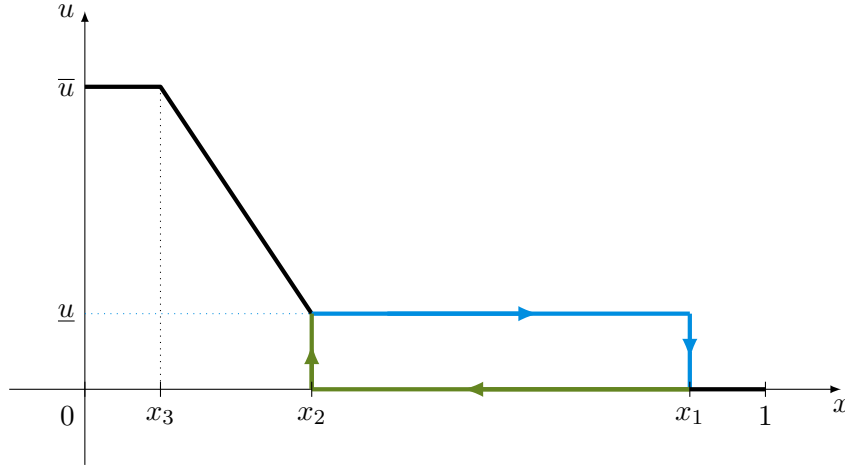
## 2.5.2 Stochastic Controllers

The performance of policies synthesized using Algorithm 2.2 is studied in this subsection. As mentioned, stochastic power management strategies performance only have a sense in the light of several utilizations.

The simulations are prepared as follows: we fix a discretization of the state space and of the random variable. By using the probabilities identified from the model of the random variable  $\xi$ , the value function is computed. Then,  $N_p$  speed profiles are generated using the same model of random variable  $\xi$ . Each one of the  $N_p$  speed profiles is called a *scenario*. A scenario corresponds to a particular realization of the *process*  $\{\xi\}$ . For each scenario, a control sequence is synthesized. The performance of the algorithm can be then analyzed using the distribution of  $N_p$  operating costs.

In order to establish a practical reference against which one can infer about the performance of the stochastic power management, we propose an empirical and intuitive power management strategy. This empirical strategy, referred to as *handmade* strategy, consists basically of setting the range extender output power inversely proportional to the SOC level in the battery. In more detail, consider three SOC reference levels  $1 < x_3 < x_2 < x_1 < 0$  and the output power bounds  $\underline{u}, \bar{u}$ .

The dependence of the control bounds on the vehicle speed is omitted to simplify the presentation of the strategy, but the dependency is taken into account in the implementation. The switching is managed by a hysteresis cycle running between  $x_2$  and  $x_1$ : the range extender is turned on if  $x_t \leq x_2$  coming from the right and it is turned off if  $x_t \geq x_1$  coming from the left. For simulation

Figure 2.3 – *Handmade power management strategy.*

purposes, the values of the parameters were set at  $x_1 = 0.8$ ,  $x_2 = 0.2$  and  $x_3 = 0.1$ . We remark that this handmade strategy is made for quantitative results comparison only and thus, the parameter setup values is not object of any particular tuning effort.

Also, in order to compare with different (and available) strategies, the deterministic open-loop controller is included in the analysis. In this case, the value function is computed for the same state space discretization  $\mathcal{Z}^\#$ , considering only the deterministic state variables, SOC, switch lock and range extender state, without using the vehicle speed as a state variable. Then, for the same set of  $N_p$  scenarios the control sequence was synthesized using Algorithm 2.3. We again stress the fact that a controller synthesized using the procedure described has no guarantee whatsoever of working, let alone incurring an acceptable performance. If the speed profile used in the computation of the value function differs from the speed profile used in the feedback, all the computation loses its meaning and one might as well have computed the value function using any different model.

The initial configuration retained for the numerical tests is a very tight one. More precisely, for the 20km driving cycle used, the vehicle starts with a low SOC (20%) so to force the optimal policy to use the range extender. The idea is that, the fuel being set to be more expensive than electric energy, an optimal strategy should use as least as possible the range extender. If the initial SOC is set at a high-level, allowing thus the vehicle to arrive to its destination without requiring the range extender, the optimal strategy clearly consists of using only battery energy. Therefore, all strategies would yield very close performance levels. By setting very tight conditions (typically, an initial SOC at 20% does not suffice to arrive at the destination) one is able to compare the performance of several different power management strategies in a more significant and non-obvious manner. The initial SOC level is set at 20%, roughly, 1% of SOC allows the vehicle to drive 1km. Simulations are run for  $N_p = 1000$  scenarios.

**Overall performance.** Figure 2.4 shows the distribution of costs obtained using two stochastic power management strategy, a deterministic power management strategy and the handmade strategy. The figure also contains the *rate of success* of each strategy, i.e., the ratio of scenarios in which the final destination was attained with a final SOC above 0%. The two stochastic strategies are referred to as A and B. Stochastic strategy A uses the probability density functions estimators obtained from the samples generated from ARMA speed profiles. In the other hand, stochastic strategy B algorithm computes the expected values considering the probability density estimators identified from the measured speed set of samples (cf. Section 7.4).

From Figure 2.4, one can verify that, although the stochastic controller B is cheaper in average than controller A, there is a drop of 20% of the success rate. This degradation of the rate of

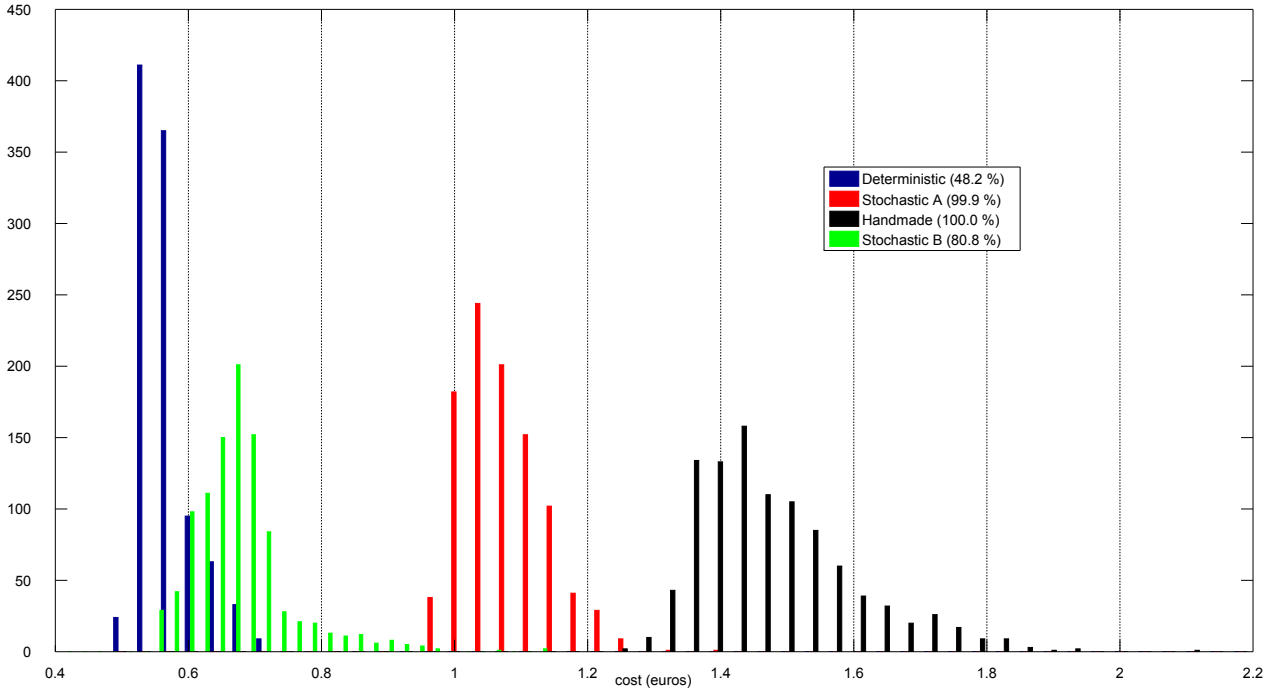


Figure 2.4 – Histogram of operating costs, in € (euros) for  $N_p = 1000$  simulations. The figure shows the costs obtained by a deterministic close-loop controller (blue), stochastic controller A (red), stochastic controller B (green) and a handmade heuristic controller (black). The legend contains the rate of success of the corresponding strategies. In absolute values, the deterministic controller has 482 successes against 999 of the stochastic controller A, 808 successes of the stochastic controller B and 1000 of the handmade controller.

success stems from the fact that, as already mentioned, for stochastic controller B, the optimization model, here including the estimators  $\hat{p}$  is different from the simulation model, i.e. simulated ARMA processes. Indeed, the stochastic processes *de facto* considered in the value function computation and the controller synthesis via the density estimators is the “real” one. However, since the stochastic process used in the numerical simulations, i.e. the ARMA(2,1) process, is of a different nature, the value function is expected to incorporate erroneous information about the probability for future realizations of  $\xi$ . This results in a greater presence of outliers<sup>5</sup>, greater standard deviation (cf. Table 2.2) and a degraded rate of success.

We draw attention to the difference in the rate of success of the deterministic close-loop strategy. With roughly 50% of rate of success, the costs presented in the figure are of little importance. The cost of a vehicle stop due to the low battery charge for a client lie well above the order of magnitude of the costs obtained by the other strategies, being estimated at around 100 €. Implemented as suggested, the simulations show that a deterministic close-loop strategy is of little or no practical interest. For the sake of argument, one can still think on how to improve the rate of success of a deterministic controller. An heuristic improvement proposed in this work consists of penalizing the value function heuristically to try to avoid a very low SOC. The results of a penalized deterministic close-loop controller is showed in Figure 2.5.

A penalization method is indeed able to bring the rate of success of a deterministic controller to a near perfect score, at any rate, comparable with the stochastic and handmade controller. After a mild effort in tuning the penalization, we observe that if one is to attain a rate of success of near 100%, the cost distribution is still somewhat higher than even a simply designed handmade controller. After these remarks, we judge the technique of using a deterministic controller unsuited for routine realistic driving profiles.

<sup>5</sup>An outlying observation is an observation that “appears to deviate markedly from other members of the sample in which it occurs” [42].

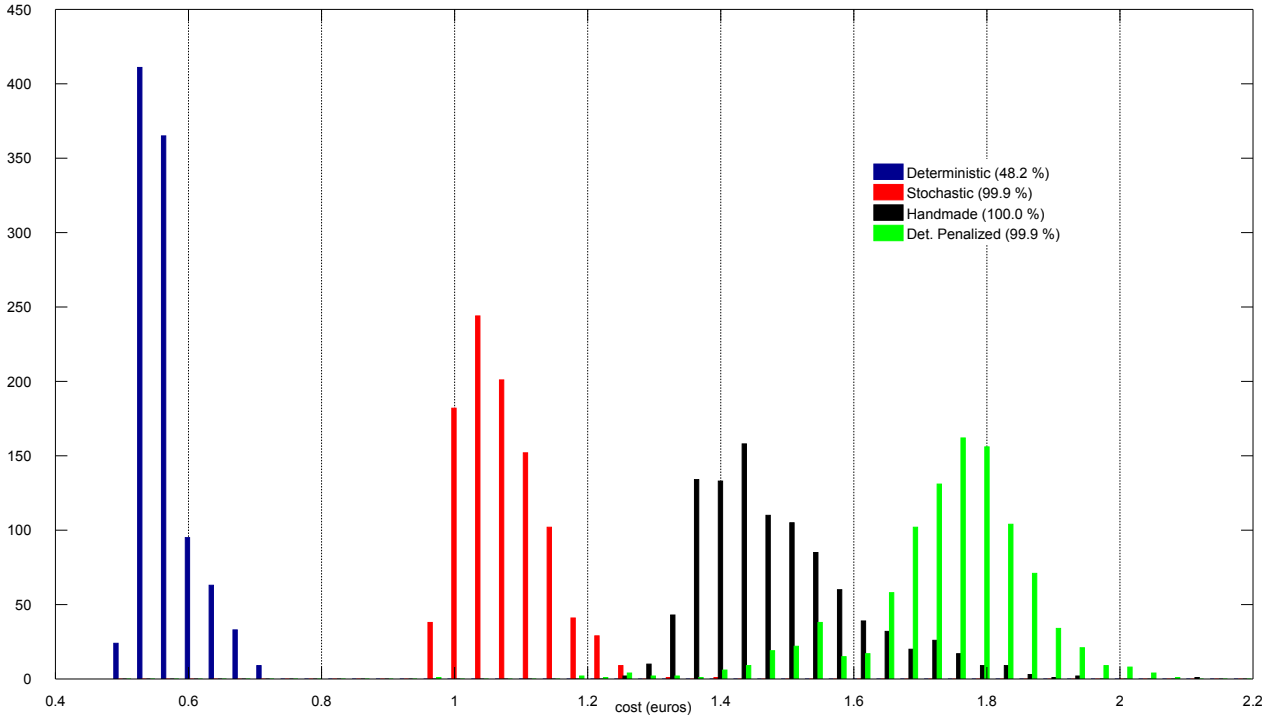


Figure 2.5 – Histogram of cost distributions including the costs of a penalized deterministic close-loop controller (green). The success rate of the penalized deterministic controller is 999 out of 1000 simulations.

Observing the information presented in Table 2.2 one can verify that the cost distribution from the stochastic controller using the adequate probability model has less outliers (11) than the other three (34 for handmade controllers and 66 when using penalized deterministic controllers). Also, when comparing the stochastic strategies, when the suitable probability model is used, we remark a somewhat lower relative number of outliers (stochastic controller A, 11 outliers) than an inadequate one (stochastic controller B, 57 outliers). The information about the number of outliers shows that the cost distribution associated to stochastic controllers is "tighter" (precisely, it has less excess kurtosis) than the other distribution, a much appreciated feature in our application. This indicates in particular that if one is to run more simulations, the new stochastic associated costs will more likely fall inside the present distribution.

Comparing the two stochastic, handmade and penalized deterministic strategies, the average cost for operating under stochastic strategies is the lowest. Also, stochastic strategies present far less variation on the total cost for the same set of speed profiles as one can observe from the standard deviation levels in Table 2.2. This behavior illustrates the robustness of the stochastic power management strategies with respect to the variations of vehicle speed. Indeed, this is one of the main reasons to consider the utilization of stochastic algorithms in the synthesis of hybrid vehicles controllers. The 0.95-quantile is the cost of operation that encompasses 95% of the cases. For the distribution of stochastic strategies, the 0.95-quantile is of 1.19€ for an average cost of 1.08€, yielding a difference of 0.08€. For the handmade and the penalized deterministic strategies, this difference is of 0.24€ and 0.17€ respectively.

**Policy structure.** Figure 2.6 compares the different SOC trajectories and hybrid controls obtained with the different algorithms tested in this work. In the first scenario, the deterministic controller enables the vehicle to reach its destination. In the second scenario, the deterministic controller fails to do so. Notice the difference of the overall control policy between the handmade and stochastic strategies. The handmade control deploys a more conservative charge sustaining operation of the RE. This particular strategy reveals itself safe, but more expensive than the stochastic controller. Indeed, given the considered parameters of the strategy, the RE output power is pro-

Algorithm	average (€)	std. deviation (€)	0.95-quantile (€)	success rate
Deterministic	0.5689	0.0392	0.6561	48.2%
Stochastic A	1.0714	0.0622	1.1912	99.9%
Stochastic B	0.6842	0.0728	0.8344	80.8%
Handmade	1.4836	0.1213	1.7290	100.0%
Det. Penalized	1.7343	0.1256	1.9028	99.9%

Algorithm	min cost (€)	max cost (€)	spread (€)	outliers
Deterministic	0.4828	0.7282	0.2454	76
Stochastic A	0.9515	1.3957	0.4442	11
Stochastic B	0.5543	1.1528	0.5985	57
Handmade	1.2555	2.2739	1.0184	34
Det. Penalized	0.9538	2.0655	1.1117	66

Table 2.2 – Table containing statistical quantities of distributions presented in Figure 2.4 and Figure 2.5.

portional to the SOC discharge. A different calibration of the parameters of the heuristic strategy (cf. Figure 2.3) might change the policy structure.

The structure of the policy of the stochastic controller is detailed in Figure 2.7. The strategy is composed of four different high-level operation phases detailed below:

- A. **Preemptive recharge phase.** The optimal stochastic controller is able to perceive the tight initial situation of the vehicle (20% of SOC for 20km). The controller operates the RE near maximum capacity to recharge the battery in this first phase of the driving cycle. Proceeding as such, the strategy increases the energy in the battery to a safe level, around 25% in this particular route. This is not an obvious optimal strategy. Indeed, recharging the battery while driving causes the output energy of the RE to pass through two conversion processes (thermal  $\rightarrow$  electric  $\rightarrow$  mechanic), instead of only one (thermal  $\rightarrow$  mechanic) as is the case if the energy is directly used to help traction. Due to inexorable losses, the energetic yield is lower. However, due to the RE power limitation and the critical charge situation, the strategy behaves in a robust less risky fashion, making a charge provision for the remainder of the trip.
- B. **Charge sustaining phase.** During this phase, the RE operates to assure a constant charge in the battery. This is an intermediary phase in which the RE operates in a more economic, but still conservative fashion.
- C. **Reactive discharge phase.** As soon as the optimal policy judges that the remaining distance can be met safely with the current SOC level, the RE output is completely halted while at the same time keeping the RE running. This can be seen as a guided discharge phase where the RE is able to deliver power at any time if needed. This proves to be the central contribution to the performance levels of the optimal stochastic controller. There is substantial fuel (and therefore cost) savings in this phase, as it lasts for most of the driving cycle. Moreover, this behavior is representative of the robustness of the stochastic controllers, as it remains ready to counter any “bad” future events.
- D. **Fuel saving phase.** Near the end of the route, the controller judges safe to turn off the RE, further reducing the cost of operating the RE.

This strategy structure is at the same time conservative, reactive and economic.

**Sensibility with respect to the initial SOC.** As mentioned in Remark 2.3, the configuration of the simulations is designed to ensure that a non-obvious RE operation is required to finish the

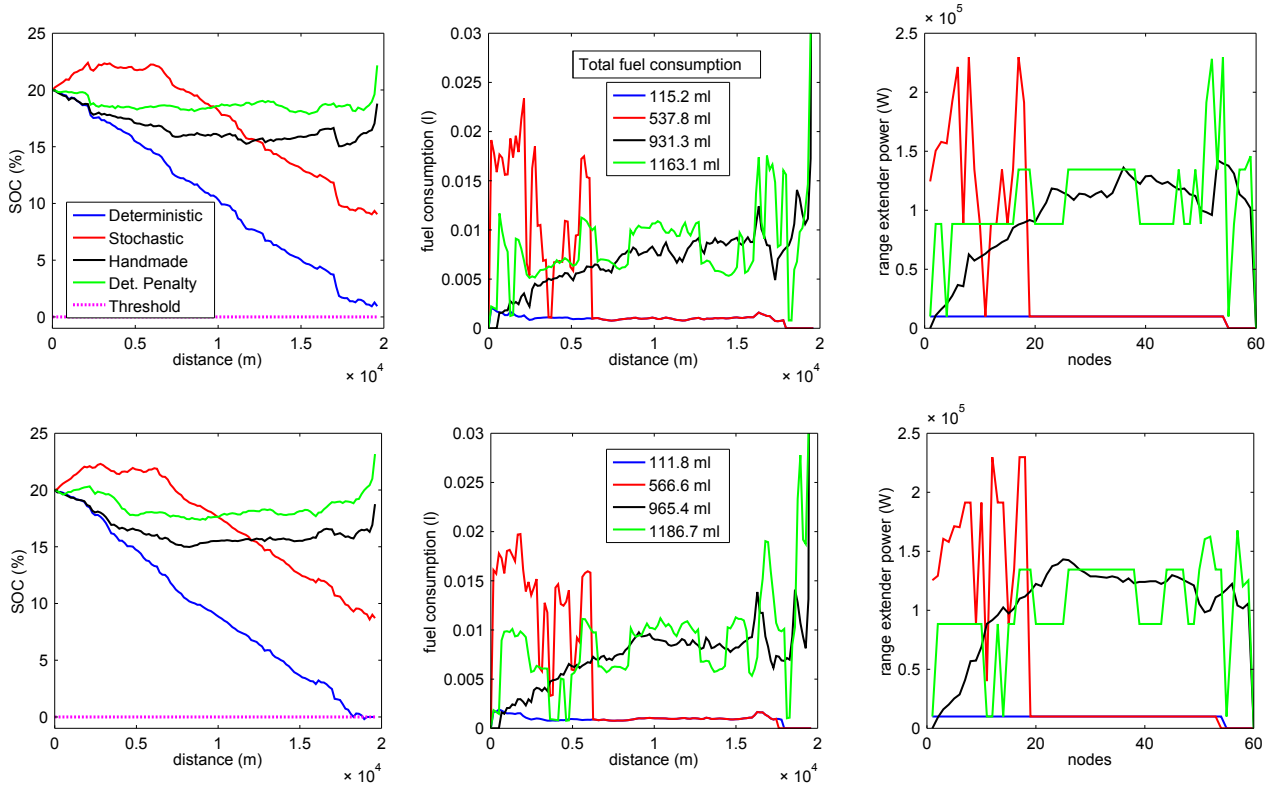


Figure 2.6 – Plots regrouping the SOC (left), instant fuel consumption (center) and range extender requested power (right) for a particular scenario realization. The curves are obtained using a deterministic close-loop controller (blue), a stochastic controller (red), a handmade controller (black) and a penalized close-loop deterministic controller (green). The center plot shows the total fuel consumption, in milliliters, for this scenario according to the strategy used.

trip. We investigate the sensibility of the strategies with respect to the initial SOC. Figure 2.8 shows the average operation costs and the success rate over 1000 simulations of different strategies. The success rate of the stochastic, handmade and penalized deterministic strategies remains near 100% for all SOC values (the stochastic and the penalized deterministic have a success rate of 99.9% and 99.8% respectively). These three strategies have a similar operation cost sensibility with respect to the initial SOC. The average cost of the stochastic strategy is the lowest between the three strategies. Indeed, the stochastic strategy average cost, for an initial SOC of 30% is 0.22€ lower than the handmade strategy, representing an improvement of near 30%. For an initial SOC of 10% this difference rises to 0.57€. In this case, the utilization of a stochastic strategy results in a relative improvement of over 35%. The deterministic strategy has a success rate of 100% only for an initial SOC above 30%. From an initial SOC below 25%, the success rate<sup>6</sup> degrades rapidly.

**Resource utilization.** The CPU running time<sup>7</sup> for computing the value function using the stochastic dynamic programming algorithm is of 1838s. The resulting data structure uses a hard disk space of over 720kb. For the computation of the value function using any deterministic algorithm using the same discretization setting, the running time is of 82s and the data structure is stored in roughly 160kB. From these values, one can observe the curse of dimensionality in effect. There is a factor of over 20 times the time of computation and a factor of over 4.5 in the hard drive space needed for storage required to add one extra state variable.

<sup>6</sup>For an initial SOC of 20% the success rate is of approximately 90%. This is a different value than the one shown in Table 2.2 because a thinner discretization grid has been used for the sensibility study.

<sup>7</sup>Intel Xeon E5504 @  $2 \times 2.00\text{GHz}$ , 2.99Gb RAM.

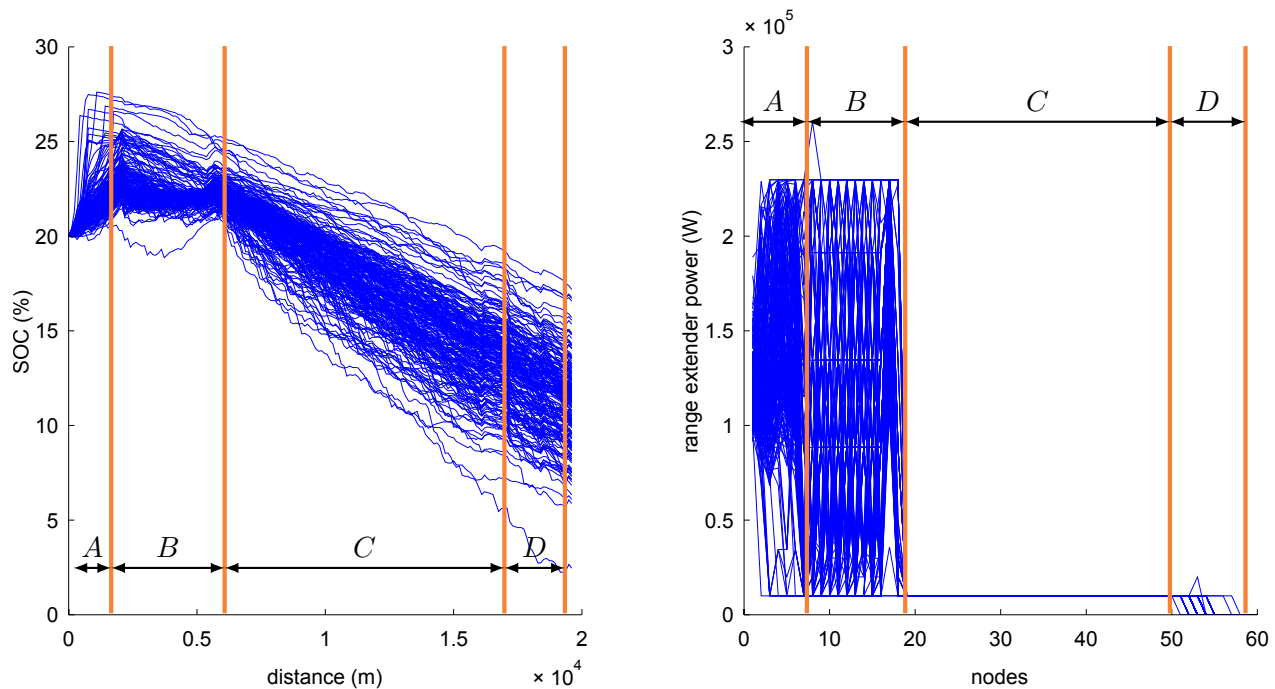


Figure 2.7 – Optimal state and control trajectories (SOC and power control) for 200 scenarios. Four distinct phases of operation appear distinctively for this route: preemptive recharge (A), charge sustaining (B), reactive discharge (C) and fuel saving (D).

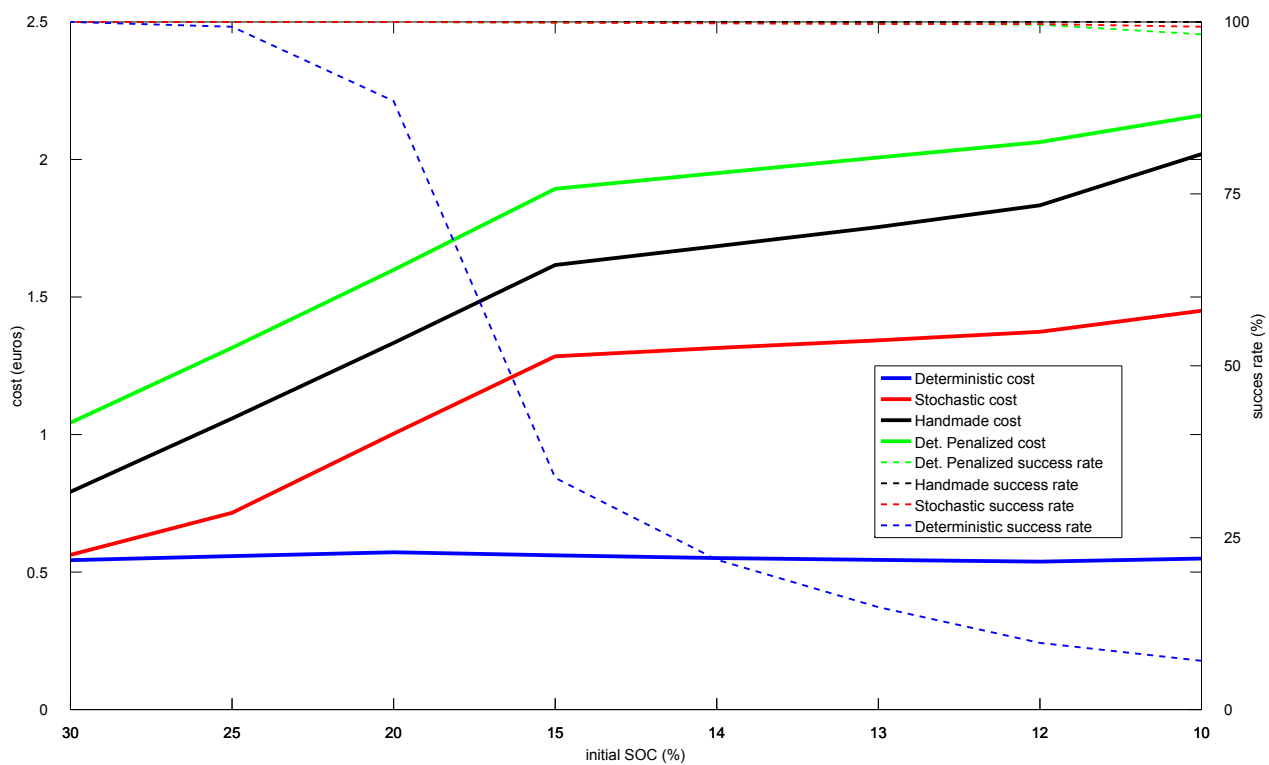






















Figure 2.8 – Strategy average cost (full lines) and success rate (dashed lines) as a function of the initial SOC over 1000 simulations. The strategies shown are the stochastic (red), handmade, (black), deterministic (blue) and penalized deterministic (green).



## 2.6 Discussion

Table 2.3 summarizes qualitative features of the different strategies tested.

	reliability	operation cost	CPU resources	overall rating
Stochastic A				
Handmade				
Stochastic B				
Det. Penalized				
Deterministic				




 = OK     
  = Acceptable     
  = KO

Table 2.3 – Table summarizing main qualitative information about the performance of simulated controllers.

In our application, the reliability is a decisive factor of the performance of a strategy. This is measured by the success rate of the strategy. We stress that the simulations consider a configuration in which it is always possible to reach the final destination. Therefore, failure to do so is judged unacceptable.

The success rate of the two stochastic strategies shows that the quality of the stochastic model used for representing the random evolution is relevant. In the best case, the stochastic process whose data is used to estimate the Markov-chain transition probabilities is the same of the process actually used in the simulations (ARMA models in our case). When the simulation model and optimization model use the same stochastic processes the success rate increased of 20% with respect to a strategy where the processes are different. In addition, the robustness of the strategy is enhanced, with a 14% lower standard deviation.

We propose two ways of harmonizing the stochastic information used in the simulation and optimization models.

- **Driving data.** A substantial database of driving profiles can be used to construct both a optimization model and a simulation platform. First, a part of the available data is used well identify the transition probabilities. Next, the part of the driving profiles not used for probability estimation are used for simulation purposes, acting as a backtesting to check how strategies would have performed if they were implemented in the past. In this approach, the quantity of good quality data must be enough to allow a statistical significance of both the probability estimator and the simulation results.
- **Modeling effort.** In the absence of a large database, an effort can be made towards the implementation of more evolved stochastic models. A more sophisticated stochastic modeling (e.g. GARCH, ARIMA, see [20]) may capture in more detail the features of the real measured driving profiles. If these models can well reproduce (stochastically) the real process underlying the measurements, they can be used in the approach proposed in this study: they are used to generate artificial data used to construct the optimization model and to simulation ends.

A heuristic strategy can be very reliable. This reliability is obtained by paying a higher cost for the operation of the RE. The stochastic feedback controller decreases the average operation cost by over 27% and over 31% if 95% of the cases are considered, when compared to the heuristic strategy considered (cf. Figure 2.3).



As it is expected, the CPU time needed to evaluate the value function in the stochastic algorithm is penalized due the increased state space. The running time is 1838s against 82s for the deterministic value function. The disk space required, in the same conditions, is of over 720kb against 160kb. In spite of the relatively higher resource needs of the stochastic controller with respect to the other controllers, the application of stochastic controllers for a real application is deemed suitable and deserves further study. We propose three utilization scenarios that might help overcome the higher computational needs and move a step towards a real hybrid vehicle stochastic controller:

- **Value function storage.** The value function is computed only once. It can then be applied successively to the optimal control on the same route. Indeed, to effectively implement stochastic strategies on-board vehicles, the value function is computed and then, it is used as a look-up table to choose the optimal control on-line. In this study, the value function is computed for a specific route. An important observation is that, once the value function is computed for a first utilization of the route, it can be used for all subsequent drives along the same route. Because the typical scenario of the REEV is a home  $\rightleftharpoons$  work drive, only a few value functions need to be stored in memory. Then, a few value functions can account for the utilization of on-line stochastic feedback control strategies in a large proportion of the vehicle mileage.
- **Trip planning.** The value function can be computed off-lined and stored outside the vehicle processors. One can think of an off-line computation of the value function as a part of a trip planning application. In this scenario, as the driver plans the trip (e.g. checking for restaurants along the route, traffic conditions, weather reports one day prior to the trip) and the route is chosen, the application computes off-line the value function. The resulting look-up table is then stored in a flash drive and uploaded to the vehicle, where it can be used to synthesize the optimal feedback control for the trip. This procedure can be of worth even in the case where the REEV is expect to perform most, namely, on a home  $\rightleftharpoons$  work scenario. The main advantage is that the computation can be made in more powerful hardware, such as smartphones, notebook computers or even desktop computers.
- **Heuristic power management design.** The stochastic strategy presents a high-level control structure that can be used to design and tune heuristic strategies. As an overall rule, heuristic strategies require little computational resources. The drawback is the tuning effort needed in the conception phase to produce a safe, robust and economic control policy. To assist with the tuning effort, the stochastic policy can offer great insight on how an optimal controller is expect to behave. This high-level control information render the task of tuning hand-tailored strategies somewhat easier. Moreover, simulating the stochastic policy to gain insight on the structure of the controller can open a path for designing more sophisticated heuristic strategies that would otherwise be very hard to calibrate or even to conceive. For instance, the heuristic controller can integrate NAV information, expected traffic conditions and driver style in its design. Therefore, regarding a real-vehicle implementation, the optimal policy structure revealed by the stochastic controller serves as a starting point for better designed heuristic strategies.

Since the main issue with the stochastic dynamic programming is the CPU requirements, we investigate the potential utilization of a different algorithm used to compute the value function, namely, the stochastic dual dynamic programming (SDDP). This investigation consisted on the construction of a suitable approximation of the vehicle model to state the optimal control problem as a convex optimization problem. Some issues concerning the implementation of the algorithm and the synthesis of the optimal controller are discussed in the next chapter.

# STOCHASTIC DUAL DYNAMIC PROGRAMMING AND POWER MANAGEMENT

---

*“Unfortunately, the world has not been designed to the convenience of mathematicians.”*

Benoît Mandelbrot

## Contents

---

<b>3.1</b>	<b>Introduction</b>	<b>61</b>
<b>3.2</b>	<b>Stochastic Dual Dynamic Programming Implementation</b>	<b>63</b>
<b>3.3</b>	<b>Feedback control synthesis</b>	<b>67</b>
<b>3.4</b>	<b>Numerical Simulations</b>	<b>68</b>
3.4.1	Linear Stochastic Logistics Model	68
3.4.2	Linear Stochastic Hybrid Vehicle Model	71
<b>3.5</b>	<b>Hybrid Vehicle Model</b>	<b>75</b>
3.5.1	Random Variables	75
3.5.2	Decision Variables	75
3.5.3	Constraints	76
3.5.4	Criterion	77
<b>3.6</b>	<b>Model Convexity Assessment</b>	<b>78</b>

---



### 3.1 Introduction

Section 2.4 studies the utilization of a stochastic dynamic programming algorithm for numerically computing the value function defined in (2.14). This stochastic dynamic programming (SDP) algorithm synthesizes an optimal feedback that controls the power split used in a REEV to minimize the expected energy cost along a given route. The SDP algorithm constructs an approximation of the value function on points of a grid obtained by discretization of the state space. Then, Equations (2.16a)-(2.16b) are solved backwards for all values of  $z$  in the grid, eventually using some interpolation whenever a value of  $z_{t+1}$  falls outside of the grid. The bottleneck of the SDP algorithm is the curse of dimensionality. This practical limitation of the size of the state dimension may be overcome by using a different form of approximating the value function. This is the main interest of the SDDP application. The objective of the study is to focus on justifying the SDDP approach to the specific hybrid vehicle model.

The SDDP algorithm was first introduced [67] as a way to find an approximate solution to stochastic optimization problems. Broadly, the idea is to approximate the expected value function by piecewise linear functions obtained by solving the associated dual optimization problem at each stage. In this context no state space discretization is needed and thus, the method is suitable for treating large dimension optimization problems. Instead, the SDDP algorithm approximates a *convex* value function by the maximum of supporting hyperplanes sequentially computed. This requires the knowledge of the value and the a sub-gradient of the function at a point. Figure 3.1 illustrates the different approximations methods of the SDP and the SDDP.

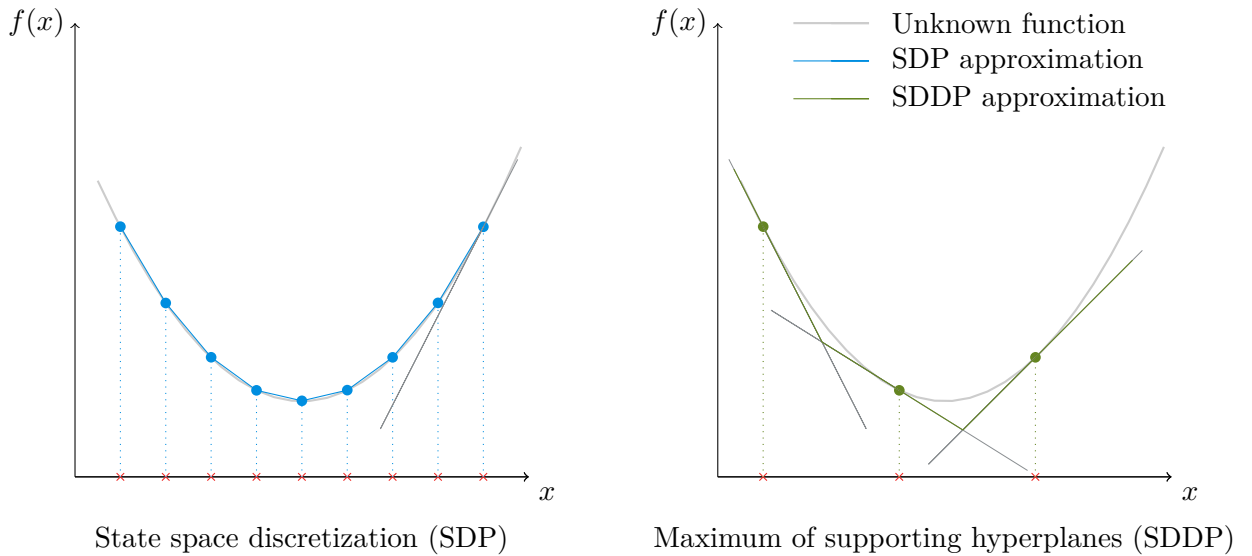


Figure 3.1 – Figure illustrating the approximations of the value function produced by a SDP (blue) and a SDDP (green) algorithm of a (unknown) function  $f$ . The SDP constructs a grid by a discretization of the state space and computes the value function on the grid. Between grid points, an interpolation method is used (here, a linear interpolation is used). The SDDP sequentially constructs supporting hyperplanes and takes their maximum as the approximation of the value function. Notice that the convexity of  $f$  is essential for the SDDP approximation.

In order to fix the main ideas, consider the following optimization problem:

$$\begin{aligned} \inf_{\substack{g_1(z_1) \leq 0 \\ g_2(z_2) \leq 0 \\ h_1(z_1) = 0 \\ h_2(z_1, z_2, \xi) = 0}} \quad & \phi(z_1) + \mathbb{E}[\psi(z_2)], \end{aligned} \quad (3.1)$$

where  $\xi$  is a random variable valued in  $\Xi$  ( $\mathbb{P}$  is a probability over the measurable space  $(\Xi, \mathcal{F})$ ),  $z_1, z_2 \in \mathbb{R}^d$  are decision variables,  $g_1, g_2 : \mathbb{R}^d \rightarrow \mathbb{R}^d$  convex constraints,  $\phi, \psi : \mathbb{R}^d \rightarrow \mathbb{R}^+$  (cost

functions) are lower semi-continuous convex functions,  $h_1 : \mathbb{R}^d \rightarrow \mathbb{R}^d$  is affine and  $h_2 : \mathbb{R}^d \times \mathbb{R}^d \times \Xi \rightarrow \mathbb{R}^d$  is affine in the second variable.

Problem (3.1) is called a *two-stage problem*. The idea is to make the first decision  $z_1$  (first-stage) before the realization of the random variable  $\xi$ . After the cost of the first decision is paid, the state  $z_2$  is obtained from decision  $z_1$  and a realization of  $\xi$  and the second stage cost is paid. In formulation (3.1) the first decision is made to minimize the *expected cost* of the decision<sup>1</sup>.

Define the optimal value of the second-stage problem function  $Q : \mathbb{R}^d \times \Xi \rightarrow \mathbb{R}^+$  as being

$$Q(z_1, \xi) := \inf_{\substack{g_2(z_2) \leq 0 \\ h_2(z_1, z_2, \xi) = 0}} \psi(z_2), \quad (3.2)$$

Define, to shorten notation, function  $\mathbf{Q}(\cdot) = \mathbb{E}[Q(\cdot, \xi)]$ . Then, problem (3.1) can be written in the form

$$\inf_{\substack{g_1(z_1) \leq 0 \\ h_1(z_1) = 0}} \phi(z_1) + \mathbf{Q}(z_1). \quad (3.3)$$

As a next step, consider the formulation of the dual problem associated to (3.2). Given  $(z_1, \xi)$ , the *Lagrangian* is

$$\mathcal{L}(z_2, \lambda, \nu; z_1, \xi) = \psi(z_2) + \langle \lambda | g_2(z_2) \rangle + \langle \nu | h_2(z_1, z_2, \xi) \rangle,$$

where we denote by  $\langle a | b \rangle$  the scalar product of  $a$  and  $b$ . The *dual function* is written as

$$L(\lambda, \nu; z_1, \xi) = \inf_{z_2 \in \mathbb{R}^d} \mathcal{L}(z_2, \lambda, \nu; z_1, \xi).$$

Defining the domain (of feasibility) of  $L$  as  $\Lambda(\cdot, \cdot) = \{(\lambda, \nu) \mid \lambda \geq 0, L(\lambda, \nu; \cdot, \cdot) > -\infty\}$ , one can state the dual problem associated to (3.2) as

$$\sup_{(\lambda, \nu) \in \Lambda(z_1, \xi)} L(\lambda, \nu; z_1, \xi). \quad (3.4)$$

The SDDP algorithm requires the knowledge of one subgradient of  $\mathbf{Q}$  with respect to the first decision variable  $z_1$ . For fixed  $(z_1, \xi)$ , if the strong duality property holds for a point  $(\lambda^*, \nu^*)$ , then  $Q(z_1, \xi) = L(\lambda^*, \nu^*; z_1, \xi)$  and, since the sub-differential and the expected value are linear operators, one obtains

$$\boldsymbol{\eta}(z_1) \in \partial \mathbf{Q}(z_1) = \mathbb{E} \left[ \partial L(\lambda^*, \nu^*; z_1, \xi) \right]. \quad (3.5)$$

This means that the solution of (3.4) allows the calculation of one subgradient  $\boldsymbol{\eta} \in \partial \mathbf{Q}$  at the point  $z_1$ . Notice that, in general,  $(\lambda^*, \nu^*)$  depends on  $(z_1, \xi)$ .

**Remark 3.1.** For a given  $(z_1, \xi)$ , if  $z_2^*$  minimizes the Lagrangian, the sub-differential with respect to  $z_1$  of  $\mathbb{E}[L]$  can be written as

$$\partial \mathbf{Q}(z_1) = \langle \nu^* | \mathbb{E} [\partial h_2(z_1, z_2^*, \xi)] \rangle.$$

Notice that  $\partial \mathbf{Q}$  does not depend on  $\lambda$ , the Lagrange multipliers associated to the inequality constraints.

<sup>1</sup>This is not the only possible formulation. As mentioned minimizing the expected cost can be justified in a context where decision  $z_1$  has to be made several times. The optimal strategy then minimizes the average cost over a large number of realizations. Otherwise, one may be interested in minimizing the *risk* of the decision or the *worst case scenario*. For a detailed discussion on the larger issues arising in stochastic programming, we refer to [78].

In the case the strong duality property does not hold, one can still obtain the sub-differential in a point on the *convex hull* of  $\mathbf{Q}$ , since  $L(\lambda^*, \nu^*; z_1, \xi) = \mathbf{conv}(\mathbf{Q}(z_1, \xi))$ .

As mentioned above, the SDDP algorithm aims at constructing a piecewise linear approximation of the function  $\mathcal{Q}$  made of supporting hyperplanes and then use it to solve the first stage problem.

More precisely, suppose that the realization set  $\Xi$  is finite. Given a feasible trial solution  $z_1^{(0)}$  (feasibility implies that  $g_1(z_1^{(0)}) \leq 0$  and  $h_1(z_1^{(0)}) = 0$ ), define the initial support plane as  $\mathcal{Q}^0 \equiv -\infty$ . Solve problem (3.2) for all  $\xi \in \Xi$  to obtain  $Q(z_1^{(0)}, \xi)$ . By solving the dual to (3.2), one obtains a subgradient  $\eta(z_1^{(0)}, \xi)$ . Now, evaluate  $\mathbf{Q}(z_1^{(0)}) = \mathbb{E}[Q(z_1^{(0)}, \xi)]$  and  $\boldsymbol{\eta}^{(0)}(z_1^{(0)}) = \mathbb{E}[\eta(z_1^{(0)}, \xi)]$  and construct a supporting plane  $\pi^{(0)}(z) = \mathbf{Q}(z_1^{(0)}) + \langle \boldsymbol{\eta}^{(0)} | z - z_1^{(0)} \rangle$ . The approximation of the expected value is updated to  $\mathcal{Q}^1(z) = \max\{\mathcal{Q}^0(z), \pi^{(0)}(z)\}$ . Then, one can solve the problem

$$\min_{\substack{g_1(z_1) \leq 0 \\ h_1(z_1) = 0}} \phi(z_1) + \mathcal{Q}^1(z)$$

and use the solution  $z_1^{(1)}$  to iterate the procedure<sup>2</sup>.

In the next section we detail the ideas exposed above.

## 3.2 Stochastic Dual Dynamic Programming Implementation

The goal in this section is to propose an implementation of a multi-stage SDDP algorithm.

The SDDP algorithm broadly consists of two stages: a *backward pass* and a *forward pass*, iterated until a stopping criteria is met. Before presenting a SDDP algorithm formulation, we discuss briefly the main concepts behind the proposed method.

The goal of solving our optimization problem is to obtain an optimal control feedback that can be applied in the “real system”. However, test the performance of controllers on the real system for development ends usually is prohibitively expensive. Such procedure requires a configurable embedded controller, a vehicle prototype and suitable testing conditions. Therefore, simulation of the real system seems to be an essential step in developing control laws before they can be tested in the real system. In the context of a SDDP algorithm, using simulation to synthesize and test control feedbacks leads to three distinct layers of approximation (the articulation of these ideas is greatly inspired from [80]).

The introduction of a (indeed any) model to simulate the behavior of the unknown quantity is a first layer of approximation. In our application, the random parameter is the vehicle future speed. The uncertain dynamic evolution of the vehicle speed is represented with an autoregressive model identified using measured samples. Then, assuming the autoregressive model is a Markov chain, discrete transition probabilities are identified using the autoregressive models. These probabilities are used to evaluate the expectation of the value function.

Solving stochastic programs requires the evaluation of the expected value of the optimal cost

$$\mathbf{Q}(z) = \mathbb{E}[Q(z, \xi)]. \quad (3.6)$$

Dismissing the very specific case of an analytical close-form for the expectation (3.6), one is eventually bound to work with discrete random variables. The expected value can then be expressed as

---

<sup>2</sup>A more complete description of the algorithm is given after below, after properly introducing the optimization problem related to the optimal power management.

a sum and problem (3.3) can be solved using the adequate mathematical programming techniques. In our context, the main issue arising is the *size* of the resulting problem (see [79] for a more complete analysis on the complexity of stochastic programs). The number of terms in the summation is exponential in the dimension of  $\xi$ . If the random vector has  $d$  independent dimensions with  $r$  values along each dimension, the total number of realizations is  $K = r^d$ . This implies that one has to solve  $K$  (deterministic) second-stage problems in order to solve the first stage problem. One can only hope to solve such problem if the second stage problems are “easy” to solve (e.g., linear programs) and the number of realizations  $K$  is not “too large”. Additionally, the two-stage problem is required to have *relatively complete recourse*, i.e., for all admissible first-stage decisions  $z \in \mathcal{Z}$  the second-stage problem has, for all realizations of the random data, a feasible solution.

A way of computing an approximation of the expect value of the value functions is the *sample average approximation* (SAA) [51, 88]. The aim of the SAA is to approach the expectation of the value function by its average value over a great number of random realizations, or *scenarios*. There are several methods for generating scenarios (see [34] for a survey). For instance, one can use a Monte Carlo-based discretization technique [81]. One then replaces the original problem of minimizing the expected value, as in criterion (3.51), with the *sample average approximation* problem (SAA) of minimizing  $\tilde{Q}(z) = \frac{1}{N} \sum_{i=1}^N Q(z, \xi^i)$  where  $\xi^i$  are  $N$  random sample points from  $\Xi$ .

In the multi-stage setting, the optimization problem can be stated as

$$\begin{aligned} \inf_{\substack{g_1(z_1) \leq 0 \\ \vdots \\ g_T(z_T) \leq 0 \\ h_1(z_1) = 0 \\ \vdots \\ h_T(z_{T-1}, z_T, \xi_{T-1}) = 0}} \mathbb{E} \left[ \sum_{t=1}^T \phi_t(z_t, \xi_{t+1}) \right], \end{aligned} \quad (3.7)$$

The complexity of a general multi-stage stochastic problem grows disastrously with the horizon  $T$ . In [79], authors argue that such problems are seemingly intractable. Even in the light of such poor prospects, particular problems may be solved using sampling methods to a reasonable accuracy. The complete recourse property continues to be required for multi-stage problems. In this context, for all stages  $t$  and all past decisions  $z_1, \dots, z_{t-1}$ , the  $t$ -stage problem has, for all realizations  $\xi_t$ , a feasible solution.

The SAA for multi-stage stochastic problems follows the same principle of the SAA of two-stage problems, only that now, the scenarios are *processes*  $(\xi_1^i, \dots, \xi_T^i)$ ,  $i = 1, \dots, N$  instead of simple realizations. The scenarios generated for the SAA approximation are structured in a *scenario tree*. A procedure for constructing scenario tree is explained further.

The *stochastic dual dynamic programming* (SDDP) is then introduced to solve the SAA problem. We next detail a SDDP algorithm implementation.

**Step 0: Scenario Tree** To construct an appropriate scenario tree, the idea is to describe successive realizations of the random variable  $(\xi_1, \dots, \xi_T)$ . At each stage  $t$ , the SAA problem considers a particular selection  $\Xi_t$  of  $\Xi$  as the support of  $\xi_t$ . (If the probability space is continuous, the  $\Xi_t$  are a particular discretization). The cardinality of each set  $\Xi_t$  is denoted by  $N_t$ . At stage  $t$ , each possible value  $\xi_t \in \Xi_t$  is called a *node*. Assuming stagewise independence between the random variables  $(\xi_1, \dots, \xi_T)$ , each node  $\xi_t$  gives origin to other  $N_{t+1}$  nodes, called *child nodes*. From this construction, one can readily notice that each node  $\xi_t$  has a unique origin node  $\xi_{t-1}$  called *parent node*. Proceeding for  $t = 1, \dots, T$ , the total number of nodes on the tree is given by  $N_1 \times \dots \times N_T$ . A *scenario* is a particular sequence  $(\xi_1, \dots, \xi_T)$  of realizations of the random variable. Figure 3.2 illustrates a scenario tree with  $T = 3$ ,  $N_1 = 1$ ,  $N_2 = 4$ ,  $N_3 = 2$ .

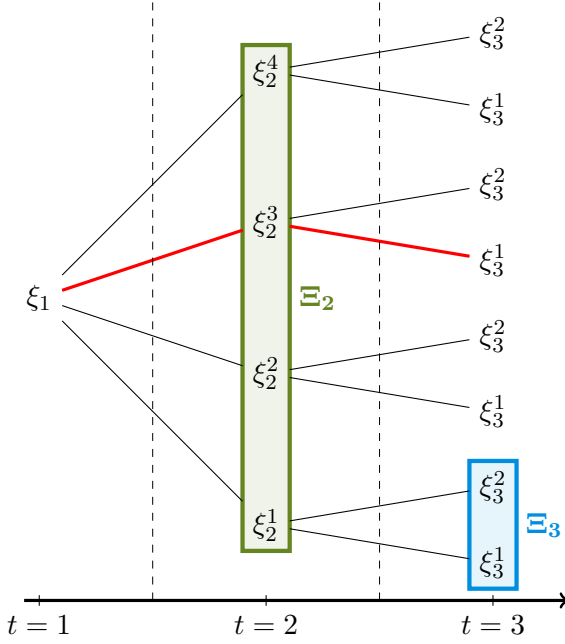


Figure 3.2 – Illustration of a scenario tree. In this case,  $T = 3$ ,  $N_1 = 1, N_2 = 4, N_3 = 2$ . Sets  $\Xi_2$  and  $\Xi_3$  are highlighted in green and blue respectively. A particular scenario realization  $(\xi_1, \xi_2^3, \xi_3^1)$  is highlighted in red.

**Step 1: Sampling of SAA scenarios** The selection (or discretization) of the probability spaces allows the SAA problem and a SDDP algorithm is applied to this approximation. Consider  $M$  iid scenarios  $(\xi_1^k, \dots, \xi_T^k)$ ,  $k = 1, \dots, M$ , sampled from the SAA scenario tree using the real probability distribution. Denote by  $Q_t$  a piecewise-linear approximation to function  $Q_t$ . It is convenient to rewrite  $Q_t$  using inequalities constraints and an auxiliary decision variable  $\theta$ . More precisely, let there be  $N_t^c > 0$  linear functions (or *cuts*)  $\pi_t^m(z) = a_t^m + b_t^m |z|$  for  $j = 1, \dots, N_t^c$ , corresponding to supporting hyperplanes. Then, one can write the condition of minimizing function  $Q_t(z) = \max_m \pi_t^m(z)$  over  $z$  as

$$\min_{z, \theta} \theta \text{ s.t. } \theta \geq \pi_t^m(z), \quad m = 1, \dots, N_t^c. \quad (3.8)$$

At  $t = 1$ , given a feasible trial decision  $\bar{z}_1$ , for each scenario  $k$  one can construct a feasible policy, i.e., a sequence  $\bar{z}_1, \dots, \bar{z}_T^k$ , by solving

$$\bar{z}_t^k \in \arg \inf_{\substack{g_t(z_t) \leq 0 \\ h_t(\bar{z}_{t-1}^k, z_t, \xi_t^k) = 0 \\ \theta \geq \pi_{t+1}^m(z), \quad m=1, \dots, N_t^c}} \phi_{t-1}(\bar{z}_{t-1}^k, \xi_t^k) + \theta. \quad (3.9)$$

The idea is to use  $\bar{z}_t^k$  as trial decisions, for some or all  $k = 1, \dots, M$ .

**Step 2: Backward recursion** The next step of the algorithm is called *backward step*. For  $t = T, \dots, 2$  the function  $\tilde{Q}_t^j(\bar{z}_{t-1}^k)$ , defined as

$$\tilde{Q}_t^j(\bar{z}_{t-1}^k) = \inf_{\substack{g_t(z_t) \leq 0 \\ h_t(\bar{z}_{t-1}^k, z_t, \xi_t^j) = 0 \\ \theta \geq \pi_{t+1}^m(z_t), \quad m=1, \dots, N_t^c}} \phi_{t-1}(\bar{z}_{t-1}^k, \xi_t^j) + \theta, \quad (3.10)$$

is solved for all child nodes  $\xi_t^j$ ,  $j = 1, \dots, N_t$  of  $\xi_{t-1}^k$ . In (3.10) we set  $\pi_{T+1}^m \equiv 0$ . Writing the dual function as

$$\begin{aligned} \mathcal{L}_t(\lambda_t, \mu_t, \nu_t; \bar{z}_{t-1}^k, \xi_t^j) &= \inf_{z_t} \phi_{t-1}(\bar{z}_{t-1}^k, \xi_t^j) + \theta + \\ &\quad < \lambda_t | g_t(z_t) > + \\ &\quad < \mu_t | \pi_{t+1}^m(z_t) - \theta > + \\ &\quad < \nu_t | h_t(\bar{z}_{t-1}^k, z_t, \xi_t^j) >, \end{aligned} \quad (3.11)$$



the dual formulation of (3.10) (c.f. (3.4)) is

$$\sup_{(\lambda_t, \mu_t, \nu_t) \in \Lambda(\bar{z}_{t-1}^k, \xi_t^j)} \mathcal{L}_t(\lambda_t, \mu_t, \nu_t; \bar{z}_{t-1}^k, \xi_t^j), \quad (3.12)$$

where the dual feasible space  $\Lambda$  is defined by

$$\Lambda(\cdot, \cdot) = \{(\lambda, \mu, \nu) \mid \lambda \geq 0, \mu \geq 0, \mathcal{L}(\lambda, \mu, \nu; \cdot, \cdot) > -\infty\}. \quad (3.13)$$

This means that, for given  $\bar{z}_{t-1}^k, \xi_t^j$ , function  $\mathcal{L}$  must be bounded below for any feasible dual variables.

Once the optimal solution  $(z_t^*, \lambda_t^*, \mu_t^*, \nu_t^*)$  of (3.12) is obtained, the algorithm then computes a sub-differential  $\tilde{\eta}_t^j(\bar{z}_{t-1}^k)$  at trial point  $\bar{z}_{t-1}^k$ . The optimal solutions  $\tilde{Q}_t^j$  and sub-differentials  $\tilde{\eta}_t^j$  are then averaged to obtain

$$\tilde{Q}_t(\bar{z}_{t-1}^k) = \frac{1}{N_t} \sum_{j=1}^{N_t} \tilde{Q}_t^j(\bar{z}_{t-1}^k), \quad (3.14)$$

$$\tilde{\eta}_t(\bar{z}_{t-1}^k) = \frac{1}{N_t} \sum_{j=1}^{N_t} \tilde{\eta}_t^j(\bar{z}_{t-1}^k), \quad (3.15)$$

and construct a new cut  $\pi_t^{N_t+1}(z_{t-1}) = \tilde{Q}_t(\bar{z}_{t-1}^k) + \tilde{\eta}_t(\bar{z}_{t-1}^k)(\bar{z}_{t-1} - z_{t-1})$  to add to the collection of existing cuts at stage  $t$ .

Notice that functions  $Q_t$  are not available for computation. Indeed, the problem one should solve is

$$\inf_{\substack{g_t(z_t) \leq 0 \\ h_t(\bar{z}_{t-1}^k, z_t, \xi_t^j) = 0}} \phi_{t-1}(\bar{z}_{t-1}^k, \xi_t^j) + Q_{t+1}(z_t). \quad (3.16)$$

However, this problem is replaced by (3.10), using the linear piece-wise approximations  $\tilde{Q}_t$  instead of  $Q_t$ . As the cuts are supporting hyperplanes of  $\tilde{Q}_t$  that may be strictly smaller than  $Q_t$ , the approximations may only be cutting hyperplanes of  $Q_t$ . Thus,  $\tilde{Q}_t \leq Q_t$  (with the inequality usually being strict [80]). As a consequence, the solution to the SAA problem using approximations  $\tilde{Q}$ ,

$$\underline{\mathcal{V}} = \inf_{\substack{g_1(z_1) \leq 0 \\ h_1(z_1) = 0}} Q_2(z_1), \quad (3.17)$$

yields a lower bound  $\underline{\mathcal{V}}$  on the optimal solution of (3.16). The lower bound  $\underline{\mathcal{V}}$  shall later be used in a stopping criteria for the algorithm.

**Step 3: Forward step** The *forward step* consists in the generation of a new set of  $M'$  scenarios sampled from the SAA tree and using the real probability distributions. The number of scenarios may well be  $M' = M$  or not. For simplicity, we shall retain that  $M' = M$  while keeping this remark in mind. (Indeed, [80] shows that the speed of convergence of the algorithm may be increased when  $M'$  changes with the iteration.) After scenarios  $(\xi_1^k, \dots, \xi_{T-1}^k)$ ,  $k = 1, \dots, M$  are sampled, one can construct feasible policies  $\bar{z}_1, \dots, \bar{z}_T^k$  that have a twofold objective. First, constructed policies can be used as trial decisions in the backward step on the next iteration. Second, they can be used to compute the total cost of each policy

$$\mathcal{V}^k = \sum_{t=1}^T \phi_t(\bar{z}_t^k). \quad (3.18)$$

The average of such over the sampled scenarii yields an (statistical, since it is sample dependent) upper bound of the optimal value of the multistage problem. More precisely, consider the sample

average and sample standard error of policies values

$$\hat{\mathcal{V}}_M = \frac{1}{M} \sum_{k=1}^M \mathcal{V}^k, \quad (3.19)$$

$$\hat{\sigma}_M = \sqrt{\frac{1}{M-1} \sum_{k=1}^M (\hat{\mathcal{V}}_M - \mathcal{V}^k)^2}. \quad (3.20)$$

By the Central Limit Theorem, the distribution of the estimator  $\hat{\mathcal{V}}_M$  approximates a normal distribution if  $M$  is arbitrarily large. Denoting  $\zeta_\alpha$  the  $1 - \alpha$  quantile of the normal distribution, one can establish a  $1 - \alpha$  confidence lower bound for  $\mathcal{V}$

$$\hat{\mathcal{V}}_M - \zeta_\alpha \frac{\hat{\sigma}_M}{\sqrt{M}}, \quad (3.21)$$

and similarly, a  $1 - \alpha$  confidence upper bound

$$\bar{\mathcal{V}} = \hat{\mathcal{V}}_M + \zeta_\alpha \frac{\hat{\sigma}_M}{\sqrt{M}}. \quad (3.22)$$

In [80] there is an argument for using the difference of upper bound (3.22) and lower bound (3.17) as a stopping criterion for the SDDP algorithm. Therefore, whenever a tolerance is required for stopping the algorithm it is understood that the stopping test checks the difference  $\Delta\mathcal{V} = \bar{\mathcal{V}} - \underline{\mathcal{V}}$ . Alternatively, one also may find interesting to stop the algorithm after a certain fixed number of iterations.

### 3.3 Feedback control synthesis

Once the algorithm stops, one can use the supporting-hyperplane approximations to synthesize a controller. The synthesis of a controller in the simulation phase can be obtained with a varying degree of approximation. First, a controller can be obtained and tested by sampling of a large number of scenarios  $M_s$  (typically larger than  $M$ ) using the scenario tree and solving the optimization problem at each stage  $t = 1, \dots, T$ :

$$\inf_{\substack{g_t(z_t) \leq 0 \\ h_t(z_{t-1}, z_t, \xi^k) = 0}} \mathcal{Q}_{t+1}(z_t).$$

For each  $k = 1, \dots, M_s$  the policy  $\bar{x}_1^k, \dots, \bar{x}_T^k$  yields a total operating cost (3.18). The average cost of  $M_s$  scenarios gives an indicator of the performance of the SDDP algorithm that engendered the approximations  $\mathcal{Q}_t$ .

Second, the performance of controllers synthesized can be studied using the procedure described in the above paragraph in more realistic conditions. Notice that in the preceding paragraph, the sampling of scenarios is made using a constructed scenario tree. We recall that the scenario tree is constructed from a discretization of the image space of the random variable (or of a particular selection if the image space is already finite) and therefore may not contain all the possible realizations of the random variable. Then, a more realistic simulation environment can be obtained if instead of sampling scenarios using a scenario tree, the random process  $\xi_1, \dots, \xi_T$  is generated using a more realistic method, e.g. non-Markovian autoregressive models.

The synthesis of a controller in “real” conditions is also implementable. This becomes the case if instead of using the dynamic equations for computing the state at the next stage one wishes to directly *measure* the state of the real system at each stage. Proceeding as such, one can use the

state feedback and past realizations of the random variable to solve sequentially first stage problems of the form (3.17). More precisely, given initial conditions  $z^*$ , solve

$$\inf_{\substack{g_1(z_1) \leq 0 \\ h_1(z_1) = 0}} Q_2(z_1)$$

to find solution  $\bar{z}_1$ . Such a solution vector contains the (feasible) control  $\bar{u}_1, \bar{w}_1$  to be applied at the first stage. In the second stage, one can measure the “real” state  $z_1^*$ , arising from a ‘true’ random realization and use it as initial condition to solve

$$\inf_{\substack{g_1(z_2) \leq 0 \\ h_1(z_2) = 0}} Q_3(z_2).$$

Notice that one keeps the first-stage constraints  $g_1, h_1$  while using the supporting hyperplane approximation for  $Q_3$ . This procedure is iterated until stage  $T$ .

In the next section, we illustrate the SDDP algorithm behavior and comment on some of its features via two numerical examples.

### 3.4 Numerical Simulations

In this section we present some numerical simulations of a SDDP algorithm. At a first moment, we use a simple stochastic logistics problem on a network. This linear model allows us to verify the convergence of the algorithm, as well as the behavior of the solution with respect to the number of stages considered. Additionally, we compare the performance of SDDP-synthesized controllers with respect to controllers synthesized using a dynamic programming algorithm.

To gain further insight on the behavior of the SDDP algorithm, we test the algorithm on a simple linear vehicle-like model.

#### 3.4.1 Linear Stochastic Logistics Model

This model describes a transportation network over which routing decisions have to be made at each stage. The random variable may be view as an uncertain traffic condition impacting the transportation costs on each route segment. The objective of the optimization is to minimize the total expected cost.

We start by presenting a 2-stage formulation. At the initial location  $A$ , one disposes of a total quantity of 1 cargo. A share of the cargo can be shipped to a northern warehouse  $W_n^1$  with a known shipping cost of  $a_n^1$  or to a southern warehouse  $W_s^2$  with a known shipping cost of  $a_s^1$ . The transport costs from those intermediate warehouses to the final destination,  $a_n^2, a_s^2$  are subject to uncertain traffic conditions, given by  $\xi$ . The random variable  $\xi$  can take two values, depending on whether there is light traffic ( $\xi = 0$ ) or heavy traffic ( $\xi = 1$ ). The traffic information becomes available only when the cargo arrives at the intermediate warehouses. At this point, the decision maker may opt to transfer a certain share of the goods from one warehouses to another, at a cost  $r_n^2$  (relocating the goods to the northern warehouse) or  $r_s^2$  (relocating the goods to the southern warehouse). This relocation and the subsequent shipment to location  $B$  are the recourse decisions and are made after the event  $\xi$  is known. The relocation costs are known with certainty and do not depend on  $\xi$ . Figure 3.3 illustrates the network for the two-stage problem. Denoting by  $x_t = (x_t^1, x_t^2, x_t^3, x_t^4)^\top$  the decision vector at stage  $t$ , the described system can modeled as

$$\min_{x_1, x_2} c_1^\top x_1 + \mathbb{E} \left[ c_2(\xi)^\top x_2 \right], \quad (3.23a)$$

under the conservation constraints

$$A_1 x_1 = b_1, \quad (3.23b)$$

$$A_2 x_1 + B x_2 = b_2, \quad (3.23c)$$

and the capacity constraints

$$x_1 \in [0, 1]^4, \quad (3.23d)$$

$$x_2 \in [0, 1]^4. \quad (3.23e)$$

The matrices in the conservation constraints are

$$\begin{aligned} A_1 &= \begin{pmatrix} 1 & 1 & 0 & 0 \end{pmatrix}, \\ A_2 &= \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}, \\ B &= \begin{pmatrix} -1 & 0 & -1 & 1 \\ 0 & -1 & 1 & -1 \end{pmatrix}, \\ b_1 &= 1, \\ b_2 &= \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \end{aligned}$$

and the cost vectors are

$$\begin{aligned} c_1 &= \begin{pmatrix} a_n^1 & a_s^1 & \infty & \infty \end{pmatrix}^\top \\ c_2(\xi) &= \begin{pmatrix} a_n^2(\xi) & a_s^2(\xi) & r_s^2 & r_n^2 \end{pmatrix}^\top. \end{aligned}$$

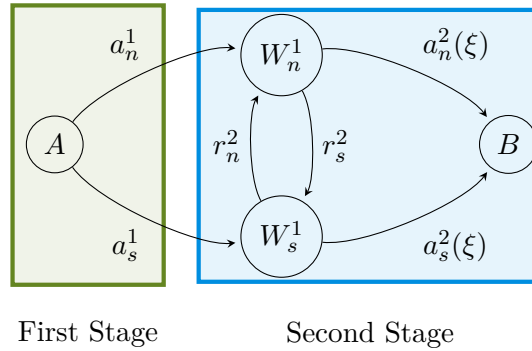


Figure 3.3 – 2-stage stochastic logistics model illustration. The objective is to transport a cargo from node  $A$  to node  $B$ , shipping the cargo, at stage  $t$ , between warehouses located in the north  $W_n^t$  or south  $W_s^t$ . The segments are named after advance cargo via north at stage  $t$  ( $a_n^t$ ), advance cargo via south at stage  $t$  ( $a_s^t$ ), relocate cargo south at stage  $t$  ( $r_s^t$ ), relocate cargo north at stage  $t$  ( $r_n^t$ ). The costs of advancing the cargo in stage 2 are impacted by uncertain traffic conditions modeled by the random variable  $\xi$ .

To study the convergence properties and overall CPU performance of the SDDP algorithm, we introduce a multi-stage logistics model with an adjustable time horizon,  $T$ . The behavior of the multi-stage case is similar to the 2-stage case already mentioned. The information of the process  $\xi = (\xi_2, \dots, \xi_T)$  enters the system sequentially before the recourse decision is made. The model associated to this case is an immediate extension of problem (3.23). Figure 3.4 illustrates the system.

In our simulations, we use the instance

- $a_n^1 = 3, a_s^1 = 2,$

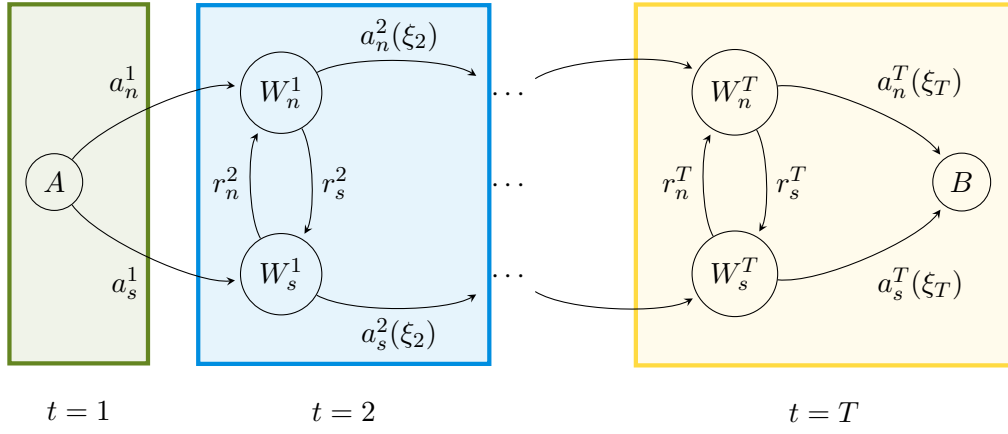


Figure 3.4 – Multi-stage stochastic logistics system illustration.

T	Solution	Iterations	CPU time (s)		
			SDP $n = 3$	SDP $n = 6$	SDDP
3	9.75	3	4.53	–	0.45
4	13.44	3	13.40	–	1.55
5	17.09	4	20.45	–	6.79
6	20.73	4	26.92	–	8.88
7	24.37	4	33.22	–	14.20
10	35.25	4	29.83	774.36	56.79
15	53.37	4	48.22	1263.93	81.33
20	71.50	4	67.16	1753.06	120.52
30	107.75	7	104.72	2620.28	1174.70

Table 3.1 – SDDP algorithm performance results on linear stochastic logistics model

- $r_s^t = 1$  and  $r_n^t = 2$  for all  $t = 2, \dots, T$ ,
- $a_n^t = 3\mathbf{1}_{\xi_t=0}(\xi_t) + 4\mathbf{1}_{\xi_t=1}(\xi_t)$  for all  $t = 2, \dots, T$ ,
- $a_s^t = 2\mathbf{1}_{\xi_t=0}(\xi_t) + 15\mathbf{1}_{\xi_t=1}(\xi_t)$  for all  $t = 2, \dots, T$ .

Here, we denote by  $\mathbf{1}_G$  the indicator function of set  $G$ .

As a first step, a value function is computed via a dynamic programming algorithm and the feedback construction, procedure described in section 3.3, is repeated  $N_f$  times to obtain a distribution of the total operation costs. The quality of the dynamic programming policies, in general, depends on the discretization step used in the construction of the discretized state space grid. In this linear problem however, since the optimal solutions are the on the boundary of the control set (bang-bang type solutions) causing the state vector also to be on the bounds of the admissible values, the discretization step does not impact the optimal solution. Since the discretization size of the state space grid used in the dynamic programming does not affect the optimal value, it remains fixed for all simulations.

Table 3.1 presents the optimal value obtained with a dynamic programming algorithm using to different state space discretization (3 and 6 points per dimension), the *average* number of iterations needed to stabilize the lower bound increase towards the optimal value and the corresponding average CPU times. Indeed, in this numerical experiment, the stopping criteria is based on the difference between the lower bound  $\underline{V}$  and the optimal value. The CPU time for the SDDP algorithms is the time spent in the construction of the supporting hyperplanes of the value function.

The upper bound  $\bar{\mathcal{V}}$  defined in (3.22) is a random variable as the cost average  $\hat{\mathcal{V}}_M$  and standard deviation  $\hat{\sigma}_M$  depend on the scenario selection at a particular iteration. Since  $\hat{\mathcal{V}}_M, \hat{\sigma}_M$  are sample estimators, to achieve a statistical significance level, the number of sampled scenarios  $M$  must be large. This is not the case in our simulations. We observe that working with  $M = 1$  provides a satisfactory rate of convergence of the lower bound  $\underline{\mathcal{V}}$  of the algorithm towards the optimal solution, the latter being given by a stochastic dynamic programming algorithm which, discretization and interpolation errors apart, guarantees the global optimum. The goal of implementing a SDDP algorithm in our application is precisely replace a dynamic programming-based solution, judged too expensive. Therefore, in a more realistic setting one cannot expect to have the optimal value available to be used as a stopping criteria. When dealing with the realistic case where no information about the optimal solution is available, one has to rely on the difference  $\Delta\mathcal{V} = \bar{\mathcal{V}} - \underline{\mathcal{V}}$  to determine the convergence of the algorithm. This in turn implies on being able to generate statistically significant upper bounds. The case  $M = 1$  is indeed a reasonable choice from the numerical performance point of view, but it does not yield statistically significant values of  $\bar{\mathcal{V}}$ . An alternative way to obtain an upper bound is to evaluate the upper bound over a moving window *over the past iterations*. More precisely, denoting  $\hat{\mathcal{V}}_M(k)$  the average cost  $\hat{\mathcal{V}}_M$  at iteration  $k$  and a window of  $I$  iterations, one may consider, at iteration  $i$ , the value

$$\hat{\mathcal{V}}_M^* = \frac{1}{I} \sum_{j=0}^{I-1} \hat{\mathcal{V}}_M^{(i-j)}, \quad (3.24)$$

and the associated sample standard deviation over the moving window  $\hat{\sigma}_M^*$ . Of course, this in turn implies that the number of iterations is large enough to justify the statistical significance of  $\hat{\mathcal{V}}_M^*$ .

In this simple application, the optimal solution is attained by the SDP with a state space discretization of only 3 points. Of course, for a finer discretization of the state space, the SDDP algorithm becomes more “competitive”. This finer discretization, however is not needed to increase the quality of the approximation of the value function. Figure 3.5 shows the operation costs obtained for several state space discretization and different stages. One can observe from the figure that there is almost influence of the number of the points of the discretized state space on the quality of the policies.

Therefore, we judge that the information in Table 3.1 does not allow for a conclusion about the CPU performance of the SDPP algorithm against a dynamic programming counterpart.

In the next section we develop a simple hybrid vehicle model to assess the quality of the SDDP approximations with respect to an optimal obtained in a closed form.

### 3.4.2 Linear Stochastic Hybrid Vehicle Model

We now introduce a simplified vehicle model. Noticeably, this model do not include the possibility of switching the RE on and off.

The state is composed of  $x$ , the SOC of the battery and  $r$ , the fuel left in the RE reservoir. The control is denoted  $u$  and represents the power produced by the RE. We shall consider that the SOC consumption may experiment a random impact, denoted by  $\xi$ . In a realistic application, this random variable may represent speed fluctuations around a recommended cruise speed given by the navigation system of the vehicle. If a stochastic model for the vehicle speed is available,  $\xi$  may be interpreted as a forecast error. Those kinds of deviations from a predicted speed impact the predicted energy consumption and are a major factor impacting several applications on the automotive industry. Besides the power management application we focus in this study, we may cite autonomy estimation for electric vehicles and routing.

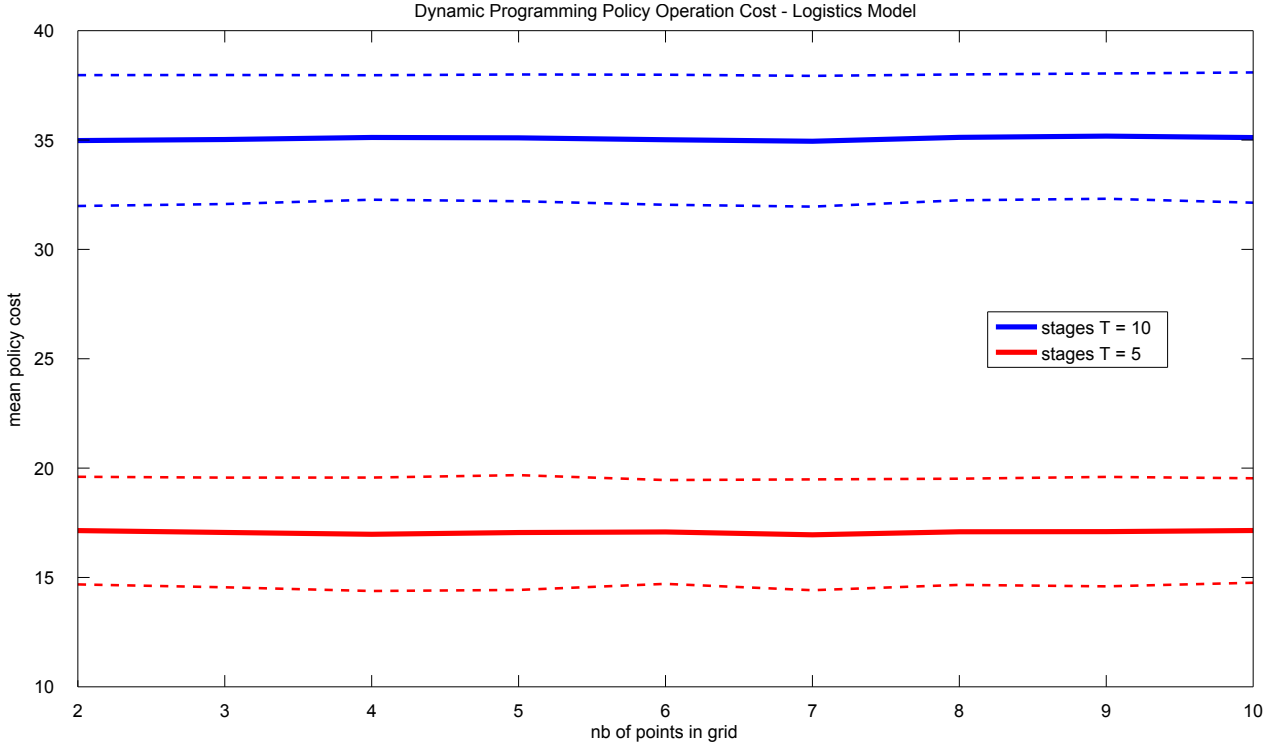


Figure 3.5 – Policy operation costs obtained using value function computed via a dynamic programming algorithm and a SDDP algorithm. The costs are function of the number of points in the discretized state space used in the dynamic programming algorithm and the number of stages  $T$ . The costs are obtained for  $N_f = 1000$  feedbacks.

The system dynamics is given by

$$x_{t+1} = x_t - a_x + u_t + \xi_{t+1}, \quad (3.25)$$

$$r_{t+1} = r_t - a_r u_t, \quad (3.26)$$

with given  $x_1, r_1$ . We assume a unit system in which all the constants needed to adjust the units accordingly are 1. In this simple model, the power produced by the RE causes the fuel level to decrease at a constant rate  $a_r > 0$ . In the other hand, producing power can reduce the rate of depletion of the battery or indeed recharge the battery. The constant  $a_x > 0$  represents an “expected” rate of depletion of the battery. The evolution of the SOC is affected by the random quantity  $\xi$  which may increase or decrease the depletion rate. We assume that capacities of the battery and reservoir are normalized, and therefore,  $x_t, r_t \in [0, 1]$  at all times. Also, the power control is bounded above by  $\bar{u}$  and thus,  $u_t \in [0, \bar{u}]$ .

The objective of a power management strategy is to minimize the expected total financial cost of a particular car trip. Denote the horizon by  $T$ , representing the total length of a given route. Indeed, REEV are expected to be equipped with navigation systems. If the driver inputs a desired geographic location, the navigation system can propose a route from its actual location to the destination point. Ergo, using the available information from the navigation system is a rather natural approach in our optimization model. In this simplified setting, the criterion one wishes to minimize is given by

$$J(z) = \mathbb{E}[(r_0 - r_T) + \beta(x_0 - x_T)]. \quad (3.27)$$

The non-negative weight factor  $\beta$  represents the relative importance of the electric energy consumption relative to the fuel consumption.

Writing the constraints  $g_t \leq 0, h_t = 0$  of the problem, one obtains

$$g_t(z_t) \equiv \begin{cases} x_t - 1 \\ -x_t \\ r_t - 1 \\ -r_t \end{cases} \quad (3.28)$$

and

$$h_1(z_1) \equiv \begin{cases} x_1 - x^* \\ r_1 - r^* \end{cases} \quad (3.29)$$

$$h_t(z_{t-1}, z_t, \xi_{t-1}) \equiv \begin{cases} x_t - x_{t-1} + a_x - u_t - \xi_t \\ r_t - r_{t-1} + a_r u_t. \end{cases} \quad (3.30)$$

The optimization problem in this case is given by

$$\min_{z \in \mathcal{Z}} \mathbb{E}[(r_0 - r_T) + \beta(x_0 - x_T)], \quad (3.31)$$

under constraints (3.28)-(3.30).

This problem allows an analytic optimal cost. Assuming  $\mathbb{E}[\xi] = 0$ , the optimal value is given by

$$J^* = (T - 1)(u^*(a_y - \beta) + \beta a_x), \quad (3.32)$$

where

$$u^* = \begin{cases} 0 & \text{if } a_y - \beta \geq 0 \\ \bar{u} & \text{otherwise.} \end{cases} \quad (3.33)$$

Indeed, one may see the term  $T\beta a_x$  as the natural loss due the battery constant depletion rate. The difference  $a_y - \beta$  balances the rate of using fuel with its price relatively to the battery usage. Since the problem is linear, one has a bang-bang solution whether using the RE at maximum power whether doing nothing and using the electric energy in the battery.

From remark 3.1 one sees that only the equality constraints  $h_t, t > 1$  contribute to the sub-differential. In this toy model, the computation can be made explicitly. Denoting  $\nu$  the multiplier associated with the equality constraints  $h_t$ , assume that, for given  $z_{t-1} := (x_{t-1}, r_{t-1}, u_{t-1})$ ,  $z_t^*, \nu^*$  optimize the associated dual problem to (3.31). From expression (3.5) and remark 3.1 one obtains

$$\partial Q(z_{t-1}) = \langle \nu^* | \mathbb{E}[\partial h_t(z_{t-1}, z_t^*, \xi_t)] \rangle,$$

which yields

$$\partial Q(z_{t-1}) = \left( 0 \ 0 \ \langle \nu^* | \begin{pmatrix} -1 \\ a_y \end{pmatrix} \rangle \right).$$

Numerical simulations are performed using the instance

- $a_x = 0.1, a_y = 1.1$ ,
- $\beta = 1, \bar{u} = 0.05$ ,
- $\xi_t$  uniformly distributed in  $[-0.01, 0.01]$ , for  $t = 1, \dots, T$ .

In this instance, the optimal value is simply

$$J^* = (T - 1)a_x. \quad (3.34)$$



In this setting, due the support of the distribution of  $\xi$ , the system can experiment “favorable” outcomes (in the sense that they tend to inflict lesser costs than the mean value of  $\xi$ ). Therefore, even with a reasonable number of cuts in each iteration, the lower bound of the SAA problem may be higher than the optimal value.

Additionally, as mentioned, the feasibility of the problem is tied to whether or not the vehicle can reach the destination given initial SOC and fuel. In the other direction, given initial SOC and fuel, one has to account for the possibility of an overcharge of the battery caused by a particular realization of  $\xi$ . The latter case can be ruled out by taking  $x^* \in [0, 0.9]$ , since  $\inf \xi_t = -0.01$  is the worst overcharge scenario. In the first case, the initial condition limits the number of possible stages. This is a natural characteristic of the system and reflects well the issues also arising in the industrial application.

Using the aforementioned instance, with  $T = 3, 4, 5, 6$ , we approximate the SAA value function by supporting hyper-planes and use them to synthesize controllers, following the procedure described in subsection 3.3. Figure 3.6 shows the distribution of total operation costs for  $M_s = 1000$  scenarios. The figure also presents the optimal cost (3.34) for each number of stages.

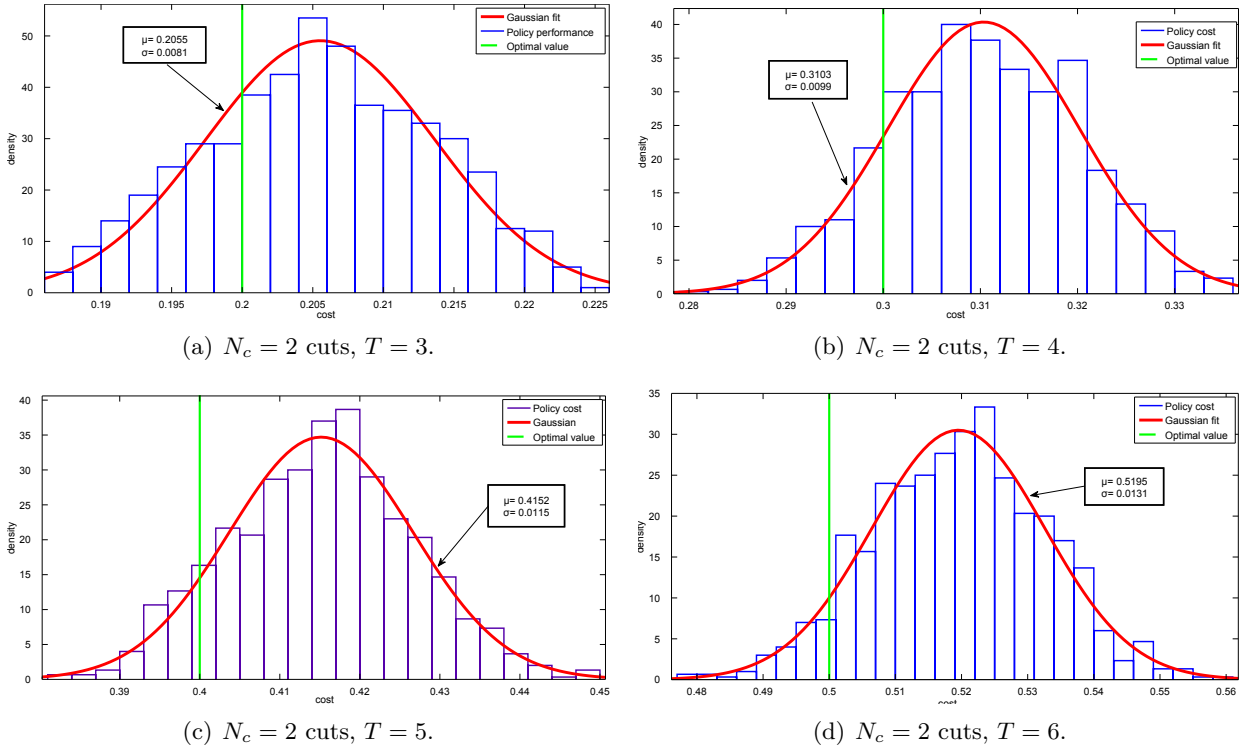


Figure 3.6 – Cost distributions for policy synthesized with SDDP applied to the SAA problem. The simulations used 1 cut per iteration and a scenario tree with  $N_t = 3$

Table 3.2 groups detailed information about the performance of the stochastic controllers obtained with the SDDP algorithm. The table contains the average cost of operation of 1000 scenarios, the loss that this average cost represents as a proportion of the optimal cost (3.34) and the deviation of the distribution as a proportion of the average cost. From this information, the SDDP feedbacks performance decreases sub-linearly as the number of stages increases. The considered loss is the distance of the average cost from the optimal cost relatively to the optimal cost. For a fixed number of cuts, the increase in the number of stages makes the approximation errors stack further, resulting in an increasing relative loss of optimality. In the other direction, the distribution of the costs deviates relatively less from the average cost as the number of stages increases. A direct consequence of this behavior is that the left tail of the distribution moves closer to the average cost at the same time that the average cost moves away from the optimal cost as the number of stages grows. The distribution tail groups the more infrequent realizations (favorable or not) of the random process. As the number of stages grows, the weight of this favorable events is less marked,

T	Average cost	% Loss	Deviation
3	0.2055	2.75	3.94 %
4	0.3103	3.43	3.19 %
5	0.4145	3.80	2.77 %
6	0.5195	3.90	2.52 %
7	0.6251	4.19	2.30 %
8	0.7296	4.23	2.06 %

Table 3.2 – *SDDP-based policies results on linear hybrid vehicle model*

since the fluctuations around the average cost diminishes as more realizations are considered in the process.

In the next section, we detail how a SDDP algorithm can be applied to a more realistic hybrid vehicle model.

## 3.5 Hybrid Vehicle Model

In this section, the main points of the vehicle model, detailed in Chapter (2) are briefly explained. We refer the reader to that chapter for the full model description. The goal is to apply the ideas described above to the optimal power management problem.

### 3.5.1 Random Variables

The random variable is denoted as  $\xi : \Omega \rightarrow \Xi$ . We shall refer to  $\xi$  as the random variable and also as its realizations, the utilization will be clear from the context. This random variable models the *vehicle speed*. Let  $\mathbb{P}$  be a probability over  $(\Xi, \mathcal{F})$ , so as to construct the probability space  $(\Xi, \mathcal{F}, \mathbb{P})$  and the process  $\{\xi\} = (\Xi, \mathcal{F}, (\xi_t)_{t \in \mathbb{N}}, \mathbb{P})$ .

### 3.5.2 Decision Variables

The decision variables are grouped into two categories, using the usual terminology of optimal control, namely, the state and control variables. The  $T$  stages correspond to points along the vehicle path, which is supposedly known in advance.

#### Control Variables

The control variables are the output power of an internal combustion engine (ICE)  $u_t$  and the order to switch the ICE's state (on or off)  $w_t$ . If  $w_t = 1$  the state of the engine should switch (if it is on, it becomes off, and vice-versa). If  $w_t = 0$  the engine's state remains the same. The control variables are grouped in a hybrid control vector  $a_t = (u_t, w_t)$ .

## State Variables

The state variables are: the state of charge (SOC) of the battery  $x_t$ , the time since the range extender was last switch  $\tau_t$ , the range extender's state (on or off)  $q_t$  and the actual vehicle speed  $y_t$ . The SOC is the image of the energy in the battery useful for driving the vehicle in nominal conditions. The state at stage  $t + 1$  is considered to be measurable with respect to the filtration engendered by the process up to time  $t$ . For that reason, at stage  $t + 1$ , whatever information about the speed process one might need is condensed in a state variable  $y_t$ . In our application we consider Markovian processes and thus,  $y_t = \xi_t$ .

The state variables are grouped in the vector  $z_t = (x_t, \tau_t, q_t)$ .

### 3.5.3 Constraints

#### Control Constraints

In the case of an ICE, the power is bounded below (in order to limit the engine vibrations) and above (in order to prevent breaking the engine). Moreover, the upper bound depends on the vehicle speed to ensure acoustic comfort inside the vehicle. Naturally, this is the case only if the ICE is on. Otherwise, the ICE output power should be zero.

Also, the ICE is subject to a decision (switching) lag. Frequent switching of the ICE is undesirable because it produces mechanical wearing off of the engine and acoustic nuisance for the driver. Decision lags impose that some minimum time is to be respected before turning the engine on and thus, avoid this issues. The decision lag period is of  $\delta = 120s$ .

These remarks, in the case of the power control, translate to the relations

$$\underline{u}q_t \leq u_t \leq \bar{u}(y_t)q_t, \quad (3.35)$$

and in the case of the switch control into

$$\begin{aligned} w_t &\in \{0, 1\} & \text{if } \tau_t = \delta, \\ w_t &= 0 & \text{if } \tau_t < \delta. \end{aligned} \quad (3.36)$$

The hybrid controls  $a_t$  are constrained to act before any realization of the random process after time  $t$ . This translates, in the Markovian case, to the measurability constraint

$$a_t \prec \xi_t. \quad (3.37)$$

#### State Constraints

The total energy of the battery is limited. Thus, the SOC levels have lower and upper bounds (one cannot recharge them indefinitely either)

$$\underline{x} \leq x_t \leq \bar{x}, \quad (3.38)$$

Also, the vehicle speed is assumed to be positive (for simplicity, bounded below by  $\underline{y} = 1$ ) and bounded above by  $\bar{y} = 140km/h$ ,

$$\underline{y} \leq y_t \leq \bar{y} \quad (3.39)$$

Switching times cannot be negative. Also, since all times  $\tau_t \geq \delta$  are equivalent, i.e. they all bring the same information about the system, one can choose any upper bound greater than  $\delta$ . Hence,

$$0 \leq \tau_t \leq \delta. \quad (3.40)$$

Finally, the on state of the range extender is represented by  $q_t = 1$ , whereas the off state corresponds to  $q_t = 0$ .

### State Dynamics

At each stage  $t$ , the SOC dynamic is given by

$$x_{t+1} = f_t(u_t, x_t, y_t, \xi_t). \quad (3.41)$$

The time since the last switch  $\tau_t$  and the range extender state  $q_t$  follow the dynamics

$$\tau_{t+1} = (1 - w_t)\tau_t + \frac{2d_t}{y_t + \xi_{t+1}}, \quad (3.42)$$

$$q_{t+1} = q_t(1 - w_t) + (1 - q_t)w_t \quad (3.43)$$

respectively.

#### 3.5.4 Criterion

We wish to minimize the *total* energy consumption in a hybrid vehicle, namely, the SOC consumption and the fuel consumed by the internal combustion engine (ICE). We shall denote the fuel consumption by

$$L_t(a_t, z_t, \xi_{t+1}) = l_t(u_t) \frac{2d_t}{y_t + \xi_{t+1}} + \rho(q_t, w_t), \quad (3.44)$$

This notation incorporates the switching cost  $\rho$ , the specific consumption  $l_t$  and the time spent on the link, assuming a linear acceleration model  $\frac{2d_t}{y_t + \xi_{t+1}}$ .

The criterion (2.12) is then written as:

$$J_1(z, a) = \mathbb{E} \left[ \sum_{t=1}^T L_t(a_t, z_t, \xi_{t+1}) + \beta(x_1 - X_{T+1}) \right]. \quad (3.45)$$

Define the function  $Q_{T+1}(z_T, \xi_{T+1})$  as the solution of the optimization problem

$$Q_{T+1}(z_T, \xi_{T+1}) := \min_{a_T \in \mathcal{A}_T} L_T(a_T, z_T, \xi_{T+1}) + \beta(x_T - x_{T+1}). \quad (3.46)$$

For  $t = T - 1, \dots, 2$ , define  $Q_{t+1}(z_t, \xi_{t+1})$  as the solution of

$$Q_{t+1}(z_t, \xi_{t+1}) := \min_{a_t \in \mathcal{A}_t} L_t(a_t, z_t, \xi_{t+1}) + \beta(x_t - x_{t+1}) + \mathbf{Q}_{t+2}(z_{t+1}), \quad (3.47)$$

where the function  $\mathbf{Q}_{t+1}(z_t)$  is defined as

$$\mathbf{Q}_{t+1}(z_t) = \mathbb{E} [Q_{t+1}(z_t, \xi_{t+1})]. \quad (3.48)$$

Then, the optimal solution of the multi-stage optimization problem (2.6) is

$$\min_{a_1 \in \mathcal{A}_1} \mathbf{Q}_2(z_1). \quad (3.49)$$

We recall that the admissible hybrid control actually depends on the current state, i.e.  $\mathcal{A}_1 := \mathcal{A}_1(z_1)$ . This is omitted to shorten the notation.

For the studied performed next, it is convenient to write the optimal control problem (2.6) as an optimization problem. To that end, denote by  $\mathbf{z} := (z, a)$  (in bold letter) the decision variables vector, regrouping the state and control variables.

With the notation of the general problem defined in (3.1), the inequality constraints are, for  $t = 1, \dots, T$ , given by

$$g_t(\mathbf{z}_t) \equiv \begin{cases} x_t - \bar{x} \\ \underline{x} - x_t \\ y_t - \bar{y} \\ \underline{y} - y_t \\ \tau_t - \delta \\ -\tau_t \\ u_t - \bar{u}(y_t)q_t \\ \underline{u}q_t - u_t, \end{cases} \quad (3.50a)$$

while the equality constraints are (given initial conditions  $x^*, \tau^*, y^*, q^*$ )

$$h_1(\mathbf{z}_1) \equiv \begin{cases} x_1 - x^* \\ \tau_1 - \tau^* \\ y_1 - y^* \\ q_1 - q^* \\ w_1(w_1 - 1) & \text{if } \tau_1 = \delta \\ w_1 & \text{if } \tau_1 < \delta, \end{cases} \quad (3.50b)$$

$$h_t(\mathbf{z}_{t-1}, \mathbf{z}_t, \xi_t) \equiv \begin{cases} x_t - f_{t-1}(u_{t-1}, x_{t-1}, y_{t-1}, q_{t-1}, \xi_t) \\ \tau_t - (1 - w_{t-1})\tau_{t-1} - \frac{2d_{t-1}}{y_{t-1} + \xi_t} \\ q_t - q_{t-1}(1 - w_{t-1}) - (1 - q_{t-1})w_{t-1} \\ w_t(w_t - 1) & \text{if } \tau_t = \delta \\ w_t & \text{if } \tau_t < \delta. \end{cases} \quad (3.50c)$$

The optimal power management decision problem can be stated as: find  $z_1, \dots, z_T$  such that

$$\inf_{\substack{g_1(\mathbf{z}_1) \leq 0 \\ \vdots \\ g_T(\mathbf{z}_T) \leq 0 \\ h_1(\mathbf{z}_1) = 0 \\ \vdots \\ h_T(\mathbf{z}_{T-1}, \mathbf{z}_T, \xi_T) = 0}} \mathbb{E} \left[ \sum_{t=1}^T \phi_t(\mathbf{z}_t, \xi_{t+1}) \right], \quad (3.51)$$

where the costs are identified as

$$\phi_t(\mathbf{z}_t, \xi_{t+1}) \equiv L_t(a_t, z_t, \xi_{t+1}) + \beta(x_t - x_{t+1}), \quad (3.52)$$

for  $t = 1, \dots, T$ .

To compute an optimal solution using a SDDP algorithm, (3.51) must be convex. Next, we detail the study on the convexity properties of the optimization problem (3.51).

### 3.6 Model Convexity Assessment

The goal of this section is to verify whether the considered optimization problem can be modeled by convex functions. If this is the case, one can apply a SDDP algorithm to solve the convex problem.

Otherwise, we shall investigate the possibilities of adapting the model to establish a satisfactory convex approximation of the original problem. Additionally, in order to use SDDP in a more effective way, we derive analytic expressions for representing the functions in the model, instead of using look-up tables.

The look-up tables present in the hybrid vehicle model are needed because some vehicle components do not have analytical expressions modeling their dynamics. The tables depend on component-specific behavior and they are constructed using cartography maps. These cartographies – for instance, mapping the engine torque and regime to a fuel consumption and the high-voltage network voltage or the battery internal resistance as a function of the battery SOC – are obtained through empirical methods and describe the behavior of the components in a certain range of input variables. As an illustration, figures 3.7 and 3.8 show the cartographies of engine efficiency as a function of engine torque and regime (the power is obtained through the product of the regime and torque) and the high-voltage network voltage as a function of the SOC.

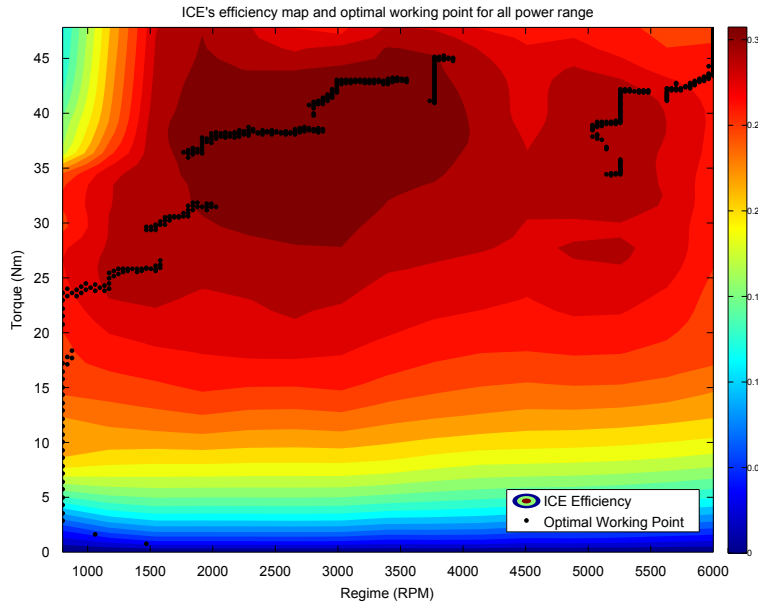


Figure 3.7 – Mapping of the ICE efficiency as a function of the ICE torque and regime. The black dots mark the optimal working points, i.e., for all iso-power hyperboles, the point that maximizes the efficiency.

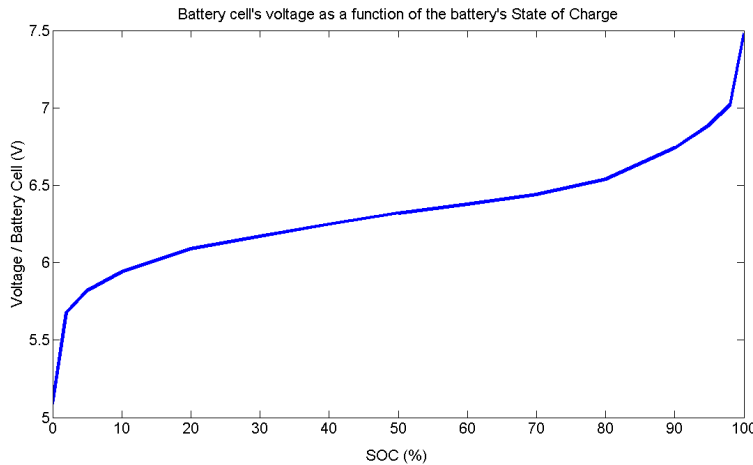


Figure 3.8 – Battery cell's voltage as a function of the battery's state of charge. A battery imposes its voltage to the high-voltage network and is composed of several cells.

We recall that the ICE has two available controls: the engine speed and its torque. These quantities form the so-called *ICE working point*. The total power is then calculated from the product of the regime and the torque. The ICE efficiency (c.f. Figure 3.7) depends on its working point. Since the ICE has, for an iso-power level, several possible working points, it is reasonable to assume that the best working point is the one that maximizes the ICE efficiency. Therefore, the ICE is seen as having only one available control, its output power  $u$ , and it is understood that the corresponding working point is set to be optimal working point, in the sense of maximizing the ICE efficiency.

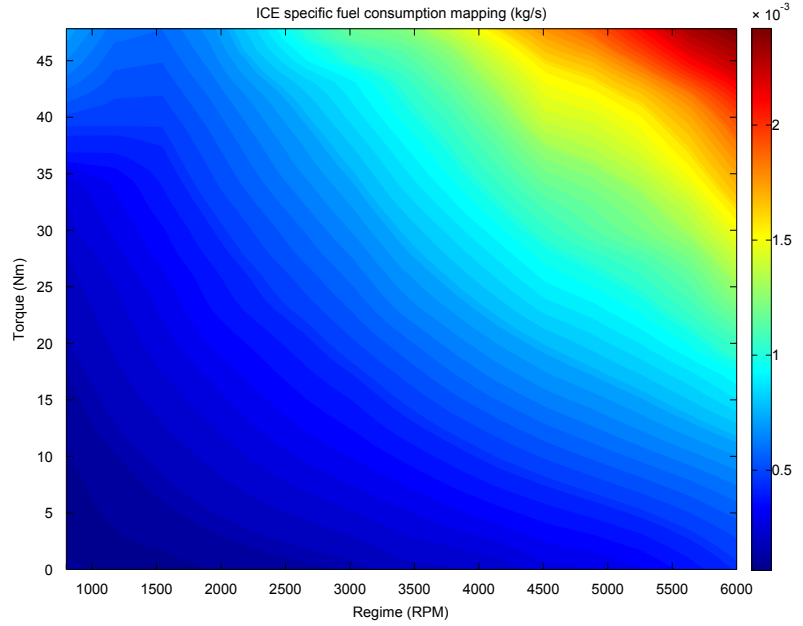


Figure 3.9 – Mapping of the ICE fuel consumption as a function of the ICE's torque and regime.

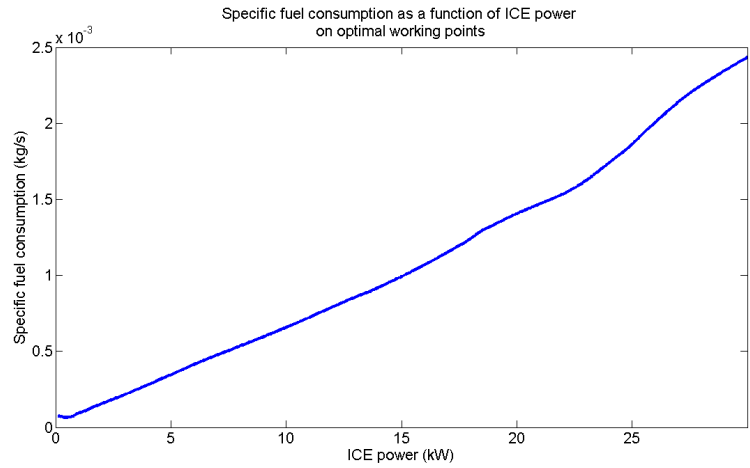


Figure 3.10 – Specific fuel consumption as a function of the ICE's power on optimal working points, i.e. working points that have maximum efficiency.

We recall the definition of convex problem. Consider the general optimization problem

$$\inf_{\substack{g(x) \leq 0 \\ h(x) = 0}} f(x). \quad (3.53)$$

Problem (3.53) is a *convex optimization problem* if three conditions are met: the objective function  $f$  is *convex*; the inequality constraints  $g$  are *convex*; the equality constraints  $h$  are *affine*. We next study the hybrid vehicle model in the light of these properties.

**Integer Variables** The first straightforward thing to notice is the presence of integer variables in the optimization problem. Indeed, the hybrid character of the formulation requires discrete state and control variables, namely, the discrete state variable  $q$  and the discrete switch control  $w$ . In our case,  $q \in Q = \{0, 1\}$  and  $w \in W = Q = \{0, 1\}$ . Since set  $\{0, 1\}$  is not convex, functions  $h_t, g_t$  are not defined over convex sets and thus, are not convex.

In view of that difficulty, we introduce a simpler but still very relevant model in which there are no discrete decisions. By considering sets  $Q^\# = W^\# = [0, 1]$  so that variables are valued in convex sets  $q \in Q^\#, w \in W^\#$ , allows for the possibility of problem (3.51) to be convex. A practical consequence of this new model is that the once-discrete state  $q$  is now interpreted as a scaling factor of the output power of the RE, ranging from 0 to 1. Recalling the expression of the SOC dynamics

described in (1.64)

$$f(u, x, y, \mathbf{q}, t, \xi) = -\frac{d(t)}{y + \xi} \frac{U_{oc}(x) - \sqrt{U_{oc}^2(x) - 4R_{bat}(x)(P_{network}(x) - u\mathbf{q})}}{R_{bat}(x)Q_{bat}} \eta(x), \quad (3.54)$$

observe that the RE state acts like an efficiency factor, *de facto* limiting the ICE output power between  $[q\underline{u}, q\bar{u}]$ . Moreover, the RE state dynamics can be rewritten simply as

$$q_{t+1} = w_t. \quad (3.55)$$

Thus, the “switch” now controls directly the modulation of output power. The direct consequence is that this new non-hybrid configuration makes the continuous power control and the “relaxed” switching redundant. Also, under these circumstances, it is not clear how to interpret the lag condition between switching. The bottom line is that this model without the integer constraints indeed allow for the possibility of a convex optimization problem suited for a SDDP algorithm. Due the striking similarity with a relaxed version of the hybrid model, we shall refer to the non-hybrid problem as the *relaxed hybrid problem*. In the new relaxed framework, the former discrete state and control lose their relevant modeling utility and become obsolete.

The above considerations lead us to opt for a relaxed version of the power management problem. In this new model, we forgo the utilization of a hybrid dynamic system modeling. The RE can be viewed as if it is running at all times. Additionally, the lag constraint is dropped. More precisely, the optimization problem now being considered is written as:

$$g_t(\mathbf{z}_t) \equiv \begin{cases} x_t - \bar{x} \\ \underline{x} - x_t \\ y_t - \bar{y} \\ \underline{y} - y_t \\ u_t - \bar{u}(y_t) \\ \underline{u} - u_t, \end{cases} \quad (3.56a)$$

$$h_1(\mathbf{z}_1) \equiv \begin{cases} x_1 - x^* \\ y_1 - y^*, \end{cases} \quad (3.56b)$$

$$h_t(\mathbf{z}_{t-1}, \mathbf{z}_t, \xi_t) \equiv x_t - f_{t-1}(u_{t-1}, x_{t-1}, y_{t-1}, \xi_t). \quad (3.56c)$$

with the cost function

$$L_t(a_t, \mathbf{z}_t, \xi_{t+1}) = l_t(u_t) \frac{2d_t}{y_t + \xi_{t+1}}. \quad (3.56d)$$

We stress that the relaxed framework result in a simplified model that is judged still very relevant for the hybrid vehicle application. Indeed, as discussed in Section 2.6, in the considered scenario, the stochastic controller consider switching off the RE only at near the end of the driving cycle. Thus, the decrease in the controller performance is expected to be low, since the optimal policy should differ from the relaxed optimal policy only at this point. As the gains from CPU time and memory usage are expected to outweigh the associate loss from dropping the possibility of switching off the RE to save fuel, the simplification is judged acceptable.

We now study more in detail the convexity of functions of relaxed problem (3.56). The fuel consumption function  $l$  and SOC dynamics  $f$  are considered to be specific quantities with respect to the length of the route link  $d_t$ . This is equivalent to fixing the link's length at  $d = 1$  km and dropping the stage subscript  $t$  dependence. Then, functions  $f$  are  $l$  are considered to satisfy  $f_t = f d_t$  and  $l_t = l d_t$  for a link of arbitrary length  $d_t$ .



**Inequality Constraints** Inequality constraints  $g_t$  are affine, with the exception of function  $\bar{u}(y)$  in (1.54) (upper bound of the ICE regime to avoid breaking). Indeed, the ICE regime has to work below a certain level that depends on the vehicle current speed  $y$ . Since we assume that the ICE always operates at optimal working points, an upper bound on the regime bound the ICE output power from above. We stress that this is a (somewhat useful) simplification of the model. One need not to consider an output power limitation because of a regime upper bound. One can always increases the torque, at constant regime, to produce an increase in the output power. However, proceeding as such degrades the ICE efficiency relatively to the optimal efficiency at a given output power. In view of the achieved simplification, we shall consider that the output power of the ICE is limited by the vehicle speed. Figure 3.11 exhibits the empirically mapped upper bound and the polynomial approximation that shall be used in this study. Because of the structure of function  $\bar{u}(y)$ , the constraint is replaced by two affine boundaries. Moreover, the affine upper bounds are to constructed to ensure that the approximations are lower than function  $\bar{u}$  for all values of  $y$ .

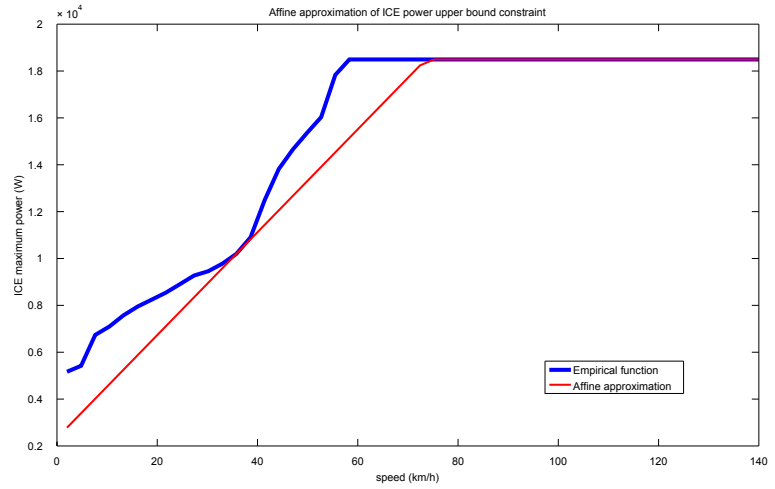


Figure 3.11 – ICE power output upper bound  $\bar{u}$  as a function of the vehicle speed.

As such, all inequality constraints in the considered model are affine, therefore convex.

**Equality Constraints** Equality constraint  $h_1$  is readily affine.

The convexity properties of equality constraints  $h_t$  are a less straightforward matter because of functions  $f_t$ .

The SOC dynamics can be seen as a variable of  $u, y$  and have two contributions from  $\xi$ : it is a function of the time spent in each link and the vehicle power demand obtained from the vehicle speed (to determine aerodynamic resistance forces and friction forces) and acceleration through Newton's (second) law. Figure 3.12 illustrates projections of  $f(\cdot, x, y, \cdot)$  for several values of  $y$ ,  $x = 20\%$ , using the available cartographies.

Notice that the feasible domain of  $\xi$  depends on  $y$ . Indeed, as the electric engine power is bounded, the vehicle cannot perform any desired acceleration from  $y$  to  $\xi$  in 1km. Thus, for instance, in the left upper-most plot ( $y = 2\text{km/h}$ ), the maximum speed achieved by the vehicle due the engine's power limitation is roughly 110km/h.

Let us now look in more detail at function  $f$ . Our goal is to establish an analytic affine approximation of  $f$ .

The expression for the SOC consumption (in %/km) is rewritten as

$$f(x, u, y, \xi) = \frac{\varepsilon(x, u, y, \xi)}{(y + \xi)RQ_{\text{bat}}} \Delta(x, u, y, \xi), \quad (3.57)$$

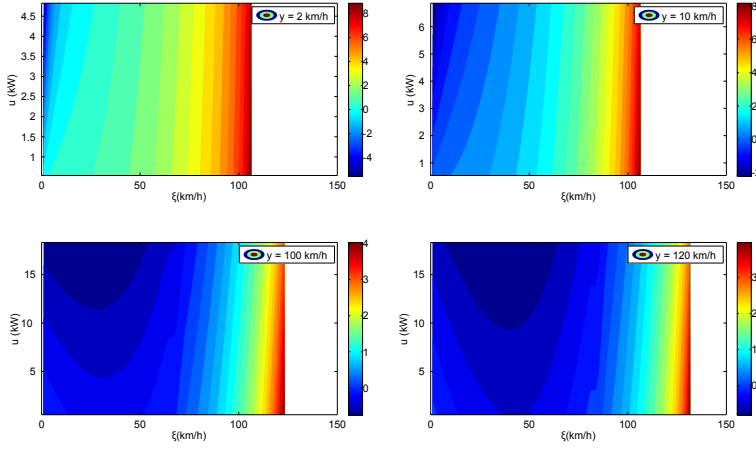


Figure 3.12 – Function  $f$  (battery energy consumption in %/km) as a function of ICE power  $u$  (at optimal working points) and future vehicle speeds ( $\xi$ ). Each contour plot is a projection on a different value of current speed  $y$ . The figures are obtained for  $x = 20\%$  (initial SOC).

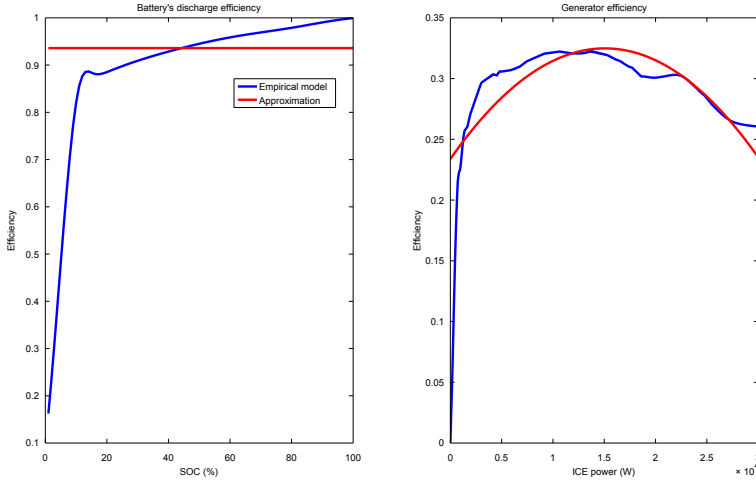


Figure 3.13 – Function and approximations of the battery's faradaic efficiency  $\varepsilon$  as a function of the SOC (left) and generator's efficiency  $\varepsilon^g$  as a function of ICE output power (right).

where

$$\Delta(x, u, y, \xi) = V(x) - \sqrt{V^2(x) - 4R(p(y, \xi) - u\varepsilon^g(u))}. \quad (3.58)$$

Equation (3.57) mainly has three terms.

The first is the battery's faradaic efficiency  $\varepsilon$ . This efficiency depends whether the battery is charging or discharging. For the discharge regime, the efficiency is a function of the battery SOC, denoted  $\varepsilon^+$ . If the SOC is increasing (battery charging), the efficiency is constant and approaches 1. Since the effective efficiency depends on the ratio of power demanded by the driver and delivered by the RE,  $\varepsilon$  is a function of the vehicle actual and future speeds and of the ICE output power, as well as of the battery SOC. Denoting by  $\mathbf{1}_A$  the indicator function of set  $A$ , the faradaic efficiency is written as

$$\varepsilon(x, u, y, \xi) = \varepsilon^+(x)\mathbf{1}_{\{u \leq p(y, \xi)\}} + \varepsilon^-\mathbf{1}_{\{u > p(y, \xi)\}}.$$

Function  $p(y, \xi)$  denotes the power needed to drive the vehicle 1 km, changing the speed linearly from speed from  $y$  to  $\xi$ . As a first approximation, we consider the faradaic efficiency  $\varepsilon \equiv 0.936$ , regardless whether the battery is charging or discharging. This allows a considerable simplification in the construction of the linear model of  $f$ .

Function  $\varepsilon^g$  is the generator efficiency. The generator efficiency is approximated by a second order polynomial. Functions  $\varepsilon^+$  and  $\varepsilon^g$  are shown in Figure 3.13 together with the constant approximations considered.

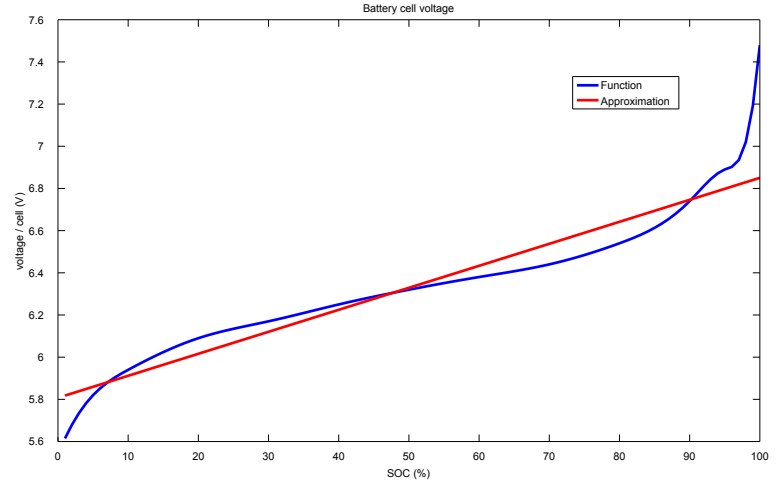


Figure 3.14 – Empirical function and approximation of battery voltage  $V(x)$ .

Using the aforementioned simplifications, the partial derivatives of  $f$  are given by

$$\frac{\partial f}{\partial x} = \frac{\partial \Delta}{\partial x} \left( \frac{\varepsilon}{(y + \xi) R Q_{\text{bat}}} \right) \quad (3.59a)$$

$$\frac{\partial f}{\partial u} = \frac{\partial \Delta}{\partial u} \left( \frac{\varepsilon}{(y + \xi) R Q_{\text{bat}}} \right) \quad (3.59b)$$

$$\frac{\partial f}{\partial y} = \left( \frac{\partial \Delta}{\partial y} + \Delta \frac{1}{(y + \xi)} \right) \left( \frac{\varepsilon}{(y + \xi) R Q_{\text{bat}}} \right) \quad (3.59c)$$

$$\frac{\partial f}{\partial \xi} = \left( \frac{\partial \Delta}{\partial \xi} + \Delta \frac{1}{(y + \xi)} \right) \left( \frac{\varepsilon}{(y + \xi) R Q_{\text{bat}}} \right) \quad (3.59d)$$

and the partial derivatives of function  $\Delta$  are given by

$$\frac{\partial \Delta}{\partial x} = \frac{dV}{dx} \left( 1 - \frac{V}{\sqrt{V^2 - 4R(p - u\varepsilon^g)}} \right) \quad (3.59e)$$

$$\frac{\partial \Delta}{\partial u} = - \left( \varepsilon^g + u \frac{d\varepsilon^g}{du} \right) \frac{2R}{\sqrt{V^2 - 4R(p - u\varepsilon^g)}} \quad (3.59f)$$

$$\frac{\partial \Delta}{\partial y} = \frac{\partial p}{\partial y} \frac{2R}{\sqrt{V^2 - 4R(p - u\varepsilon^g)}} \quad (3.59g)$$

$$\frac{\partial \Delta}{\partial \xi} = \frac{\partial p}{\partial \xi} \frac{2R}{\sqrt{V^2 - 4R(p - u\varepsilon^g)}}. \quad (3.59h)$$

Abbreviating  $\zeta = (x, u, y, \xi)$ , the linear model  $\hat{f}$  around a point  $\zeta^* = (x^*, u^*, y^*, \xi^*)$  is given by

$$\hat{f}(\zeta) = f(\zeta^*) + (x - x^*) \frac{\partial f}{\partial x}(\zeta^*) + (u - u^*) \frac{\partial f}{\partial u}(\zeta^*) + (y - y^*) \frac{\partial f}{\partial y}(\zeta^*) + (\xi - \xi^*) \frac{\partial f}{\partial \xi}(\zeta^*). \quad (3.60)$$

Now we proceed to construct polynomial (analytic) expression for  $V, p$  to be inserted in the partial derivatives (3.59)

Figures 3.14 and 3.15 shows the original function (obtained using a mapped cartography) and a polynomial model that approximates the data.

In the case of cell voltage, the polynomial approximation considered is a linear function. The vehicle power demand  $p$  is approximated by a second degree polynomial in both variables. The

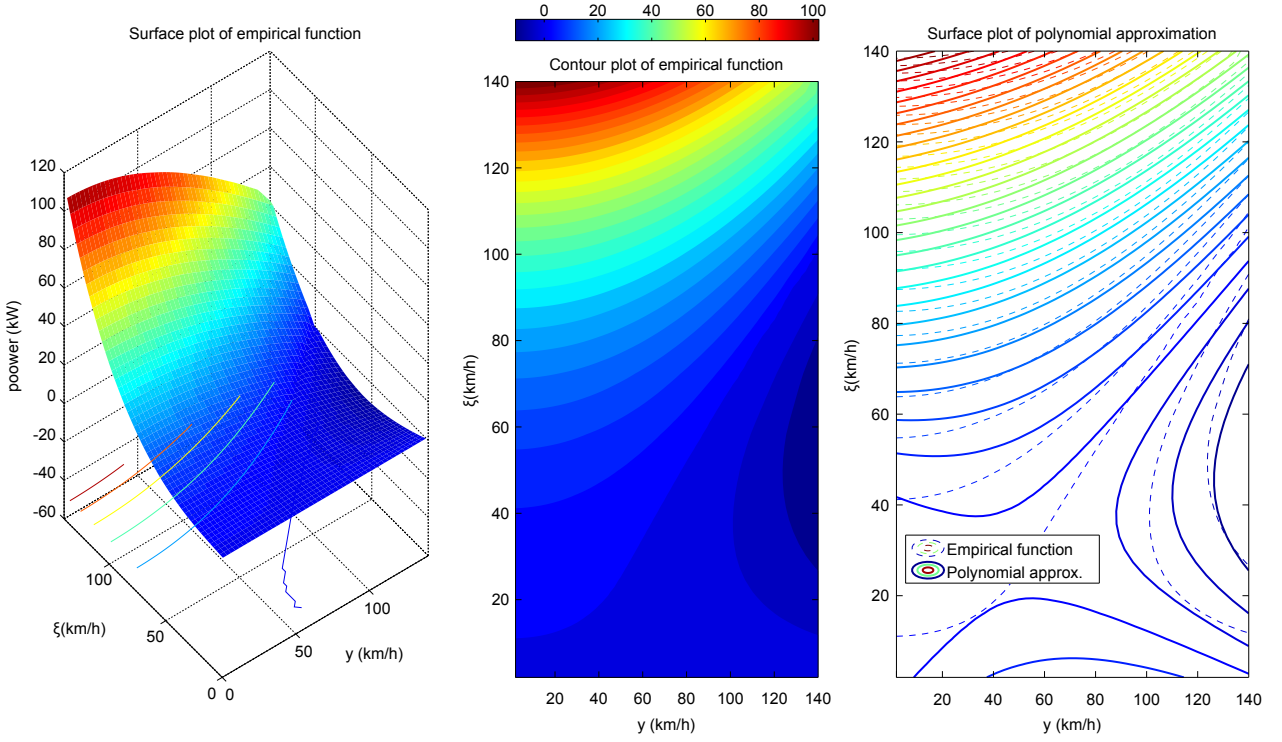


Figure 3.15 – Function and polynomial approximation of function  $p(y, \xi)$ .

analytic expressions of  $\Delta$  are then used to obtain an analytic expression that approximates  $f$  linearly at a given point  $(x^*, u^*, y^*, \xi^*)$ .

Figure 3.16 compares the behavior of function  $f$  obtained using (3.59) against the empirical data. The figure shows two different affine model, obtained centering the approximation of the empirical model around two different points. The points are  $\zeta_1^* = (0.3, 9.6, 40, 50)$  and  $\zeta_2^* = (0.5, 17.17, 30, 50)$ , where the SOC is valued between 0 and 1 and the remaining units are, respectively, kW and km/h. The consideration of these points is because the state of the vehicle is most likely to operate around those points of SOC, ICE output power and speeds.

**Remark 3.2.** Figure 3.16 shows that the empirical model for the SOC consumption is quasi-convex, and good indications that it is indeed convex. In particular, the model is monotonically decreasing with respect to the ICE output power. We recall that although there is some evidence that  $f$  is convex, the property cannot be fully asserted only by looking at the contour curves of  $f$ .

**Remark 3.3.** By plugging explicitly the structure of function  $h_t$  (3.56c) into (3.11), the dual function becomes

$$\begin{aligned} \mathcal{L}_t(\lambda_t, \mu_t, \nu_t; \bar{z}_{t-1}^k, \xi_t^j) &= \phi_{t-1}(\bar{z}_{t-1}^k, \xi_t^j) + \theta + \langle \nu_t | -f_{t-1}(\bar{z}_{t-1}^k, \xi_t^j) \rangle + \\ &\inf_{z_t} \left\{ \langle \mu_t | \pi_{t+1}^m(z_t) - \theta \rangle + \langle \lambda_t | g_t(z_t) \rangle + \langle \nu_t | x_t \rangle \right\}. \end{aligned}$$

This consideration is important for computational efficiency, since the minimization in  $z_t$  does not depend on the random variable. This feature allows for cuts to be “transported” from (3.10) to (3.10) using  $\xi_t^i, i \neq j$ .

It is expected that, using the constructed approximated affine model for function  $f$ , the control feedbacks synthesized by an SDDP algorithm should behave well when applied to the empirical model. In conclusion, the affine model introduced ensures that the equality constraints are linear.

**Criteria** For the convexity of problem (3.56), it is required that functions  $L_t$  are convex. Convexity of  $L_t$  depends on convexity properties of functions  $l_t$ , namely, the ICE fuel consumption. Figure

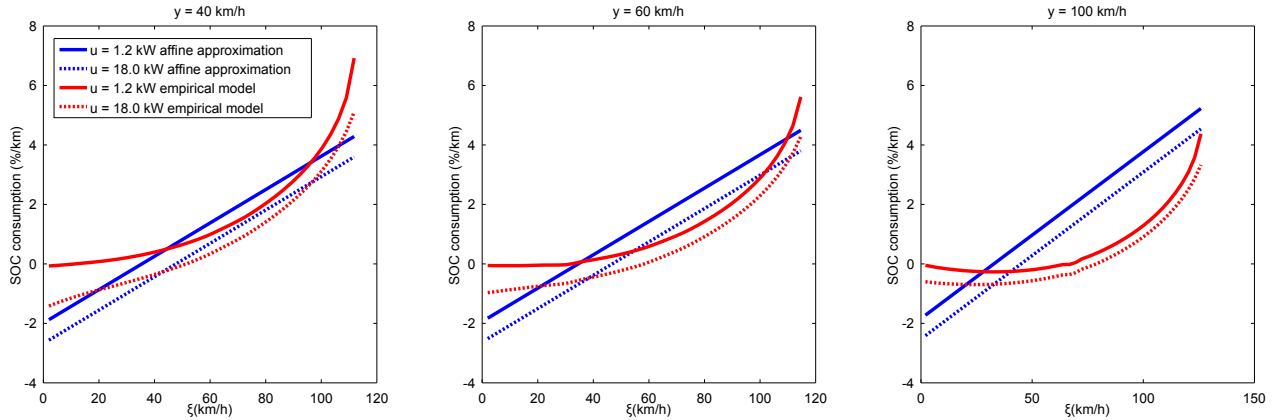
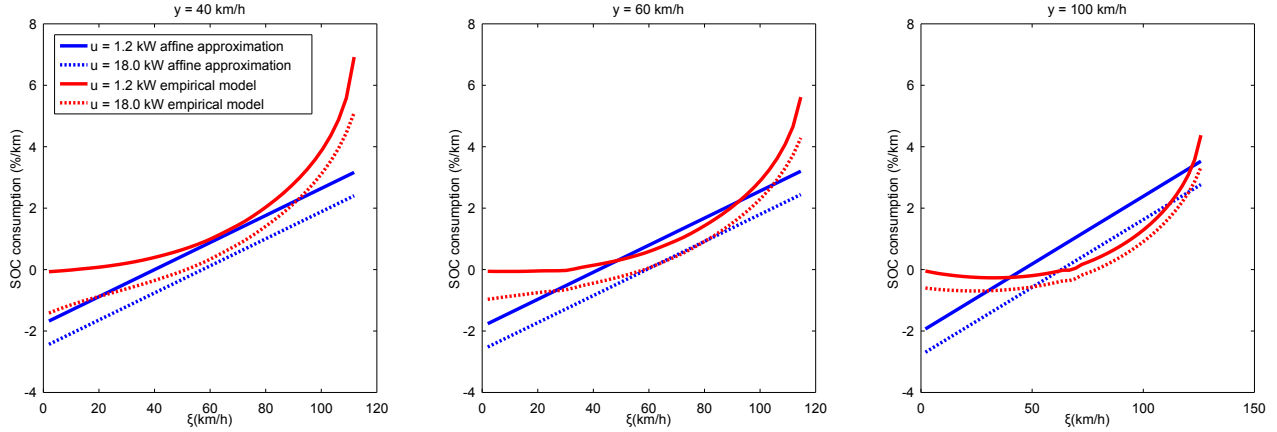


Figure 3.16 – Function  $f$  (battery energy consumption in %/km) as a function of future vehicle speed  $\xi$  for two values of ICE power  $u$  (on optimal working points). The curves show the mapped cartography functions (blue) against the affine approximation of function  $f$  (red) centered at different points.

3.10 shows the specific fuel consumption (in units of kg/s) as a function of the ICE power, obtained from the optimal working points. Qualitatively speaking, by observing Figure 3.10 one can assess that, although the specific consumption curve is not convex it is not far from it. Figure 3.17 shows one affine approximation obtained for the specific consumption and the empirical function.

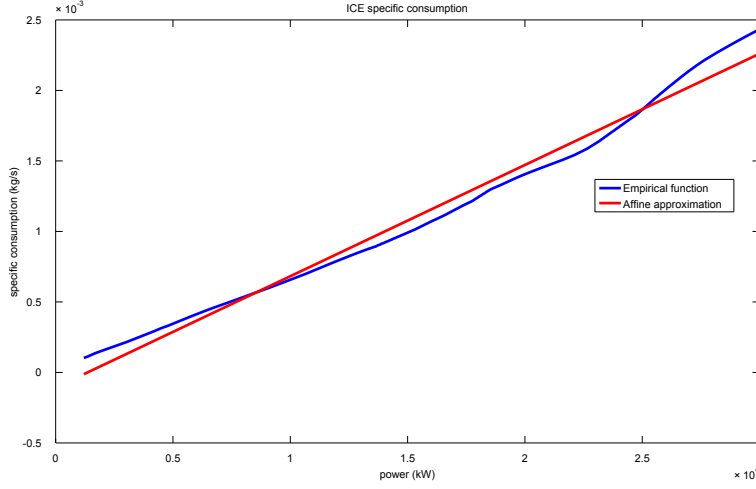


Figure 3.17 – Empirical function (blue) and affine approximation (red) of specific fuel consumption  $l(u)$ .

Figure 3.18 reports the level curves of function  $L(z, a, \cdot)$  as function of the future speed  $\xi$ .

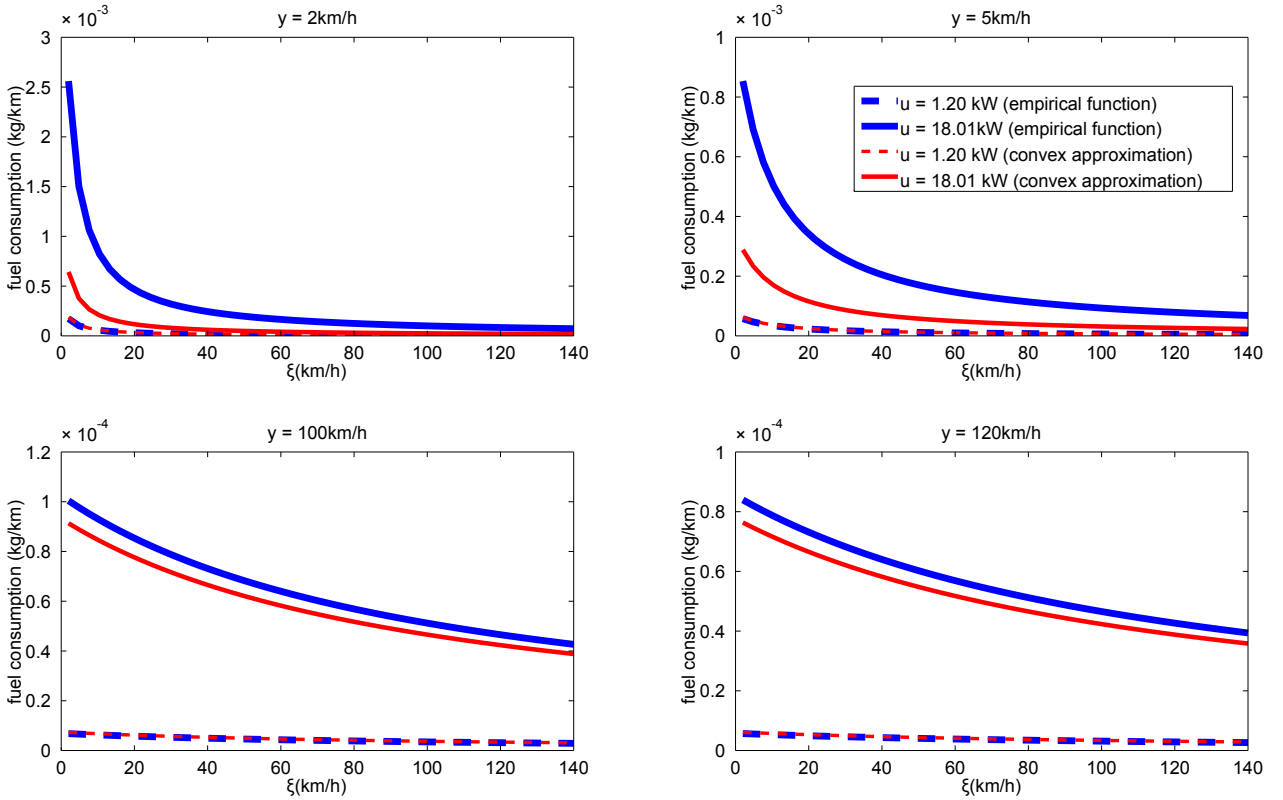


Figure 3.18 – Criteria  $L$  (fuel consumption in kg/km) as a function of future vehicle speed for two values of ICE power  $u$  (on optimal working points). The curves show the mapped cartography functions (blue) against the polynomial approximation of criteria  $L$  (red).

From the cuts shown in Figure 3.18 one can infer that function  $L$  is quasi-convex in and, qualitatively, that function  $l$  is convex. Indeed, the function appears to be monotonically decreasing in all variables.

In conclusion, qualitatively, one can verify that the approximations of the inequality constraints and the criteria are convex functions. Together with the affine approximations of the inequality and

equality constraints, the relaxed problem should prove to be a useful model for a first implementation of the SDDP algorithm application to the optimal power management synthesis. As already mentioned, the interest on such approach is enabling a possible faster way of computing optimal power management policies.

# CHANCE CONSTRAINED POWER MANAGEMENT STRATEGIES

---

*“I must complain the cards are ill shuffled till I have a good hand.”*

Jonathan Swift

## Contents

---

4.1	Introduction . . . . .	91
4.2	Probability Constraints . . . . .	91
4.3	Dynamic Programming . . . . .	95
4.4	Chance-constrained Power Management . . . . .	99
4.5	Discussion . . . . .	103

---





## 4.1 Introduction

This chapter seeks to provide an adequate mathematical framework for the synthesis of stochastic power management strategies incorporating a *chance constraint*. A chance constraint, or probability constraint, is introduced to limit the rate of failure of a stochastic policy.

More precisely, we wish to replace *almost-sure* (a.s.) constraints of the kind

$$X_t \geq 0, \quad t = 1, \dots, T, \quad (4.1)$$

with  $T > 0$ , for constraints of the kind

$$\mathbb{P}(X_1 \geq 0, \dots, X_T \geq 0) \geq \alpha, \quad t = 1, \dots, T, \quad (4.2)$$

where  $\mathbb{P}$  is a probability and  $\alpha \in [0, 1]$  is a *confidence level*. The almost-sure constraints stress the fact that the constraints must be satisfied for almost all realizations of the random variables. In general, the drawback in considering almost-sure constraints in stochastic optimization is that they tend to produce conservative feedbacks, sometimes leading to a prohibitive cost of operation. In broad terms, when the control policy is synthesized using a chance constraint instead of an a.s. constraint, scenarios which were before not tolerated because they violate the hard constraint are now admissible. The associated confidence level has the role of limiting the frequency at which those scenarios can occur.

The chance constrained model is expected to add more flexibility to the feedback control synthesis. The optimization problem stated under this kind of constraint has a greater admissible search space for the optimal policy. Therefore, the optimal cost is no bigger than the cost using a tighter search space.

This chapter is organized as follows: Section 4.2 presents general remarks about chance constrained problems. This section introduces a chance constrained optimization problem in a slightly more general formulation than that of the optimal stochastic power management, presented in Chapter 2. In addition, the power management problem formulation is adapted to better allow the study of chance constraints. Section 4.3 presents a different, but equivalent, formulation of the optimization problem that can be solved more readily via a dynamic programming algorithm than the original formulation. Also, we detail the dynamic programming principle for the equivalent formulation in a form that can be readily used for the numerical synthesis of optimal feedback using a dynamic programming algorithm.

## 4.2 Probability Constraints

In this section we discuss the relevance of introducing probability constraints in stochastic optimization problems.

When facing the problem of decision making under uncertainty one has to take into consideration several aspects of the decision making process. This step is necessary to formulate an adequate problem (in mathematical terms) using an adequate model of the system and its environment. The problem can then be solved and the solution, if any, may be used for analysis or operating purposes.

The first important aspect of the decision making process is to realize how the decision and observation of the uncertainty parameters are related. The first case is when only one decision is made and then a subsequent observation is possible. Decisions of this kind are called *open-loop decisions*. The second case regards decision processes in which after a decision and a observation, another decision follows. The decision in hindsight is labeled *recourse decision*. In this study we

are interested in dynamic multi-stage stochastic optimization problems, which translates to several recourse decisions as the random process unfolds.

We remark that the chance constraint optimization problem is analyzed in a more general formulation than the one required for the application to the optimal power management application.

In mathematical terms, the general class of stochastic optimal control problem considered in this work may be formulated as

$$\min_{\substack{\mathbf{X}=(X_{t_0},\dots,X_T) \\ \mathbf{U}=(U_{t_0},\dots,U_{T-1})}} J(\mathbf{X}, \mathbf{U}), \quad (4.3a)$$

subject to

$$\mathbf{X} \in \mathcal{X}, \quad (4.3b)$$

$$\mathbf{U} \in \mathcal{U}(\mathbf{X}), \quad (4.3c)$$

$$h(\mathbf{X}, \mathbf{U}, \xi) = 0, \quad (4.3d)$$

$$g(\mathbf{X}) \leq 0, \quad (4.3e)$$

$$U_t \preceq \mathcal{F}_t. \quad (4.3f)$$

Here, we consider that  $K$  decisions  $U_t$  have to be made at each stage  $t = t_0, \dots, t_0 + K - 1 \triangleq T - 1$  in order to control the state of the system  $X_t$  such that the cost of operation  $J : \mathcal{X} \times \mathcal{U} \rightarrow \mathbb{R}^+$  is minimized. Notice that the system starts at time  $t_0$  and the problem has  $K + 1$  stages. The state process  $\mathbf{X}$  is valued in  $\mathcal{X} \triangleq \mathbb{X}^{K+1}$ , where  $\mathbb{X} \subset \mathbb{R}^n$ . Similarly, the control process  $\mathbf{U}$  is valued in  $\mathcal{U}(\mathbf{X})$ . The notation  $\mathcal{U}(\mathbf{X})$  indicates that, for each stages, the control set may depend on the value of the state, i.e.,  $U_t \in \mathcal{U}(X_t) \subset \mathbb{R}^m$ . The random process is denoted  $\xi \triangleq (\xi_t)_{t_0 \leq t \leq T}$ , with each  $\xi_t : \Omega \rightarrow \Xi$ . We denote  $\mathcal{F}_t$  the  $\sigma$ -algebra generated by the process  $(\xi_{t_0}, \dots, \xi_t)$  and  $\mathbb{P}_t : \mathcal{F}_t \rightarrow [0, 1]$  a probability. The equality constraints  $h : \mathcal{X} \times \mathcal{U} \times \Xi^{K+1} \rightarrow \mathbb{R}^{m_h}$  may refer to some dynamic equations dictating the evolution of the state variables, relating thus  $X_{t+1}$  with  $X_t, U_t$  and  $\xi$ . They usually are given by physical models inherent to the nature of the system considered. The inequality constraints  $g : \mathcal{X} \rightarrow \mathbb{R}^{m_g}$  represent quantities that one has to keep above or below a certain level throughout the operation of the system. Constraints (4.3b)–(4.3e) have to be satisfied at each stage with probability 1. The information (or measurability, or filtration) constraints (4.3f) are functional and non-pointwise (except for trivial cases  $\mathcal{F}_t \equiv \mathcal{F}_{t_0} \equiv \{\emptyset, \Xi\}$ ). This qualitative difference introduces a deal of theoretical and computational challenges [45].

If some realizations of the random data causes the failure of the system (where the concept of failure depends on the context), one may find interesting to make decisions that preempt the occurrence of such realizations. In reference to problem (4.3), a failure is considered to be a violation of the inequality constraints (4.3e). Indeed, constraints (4.3b)–(4.3d) are physical considerations as part of the system model while (4.3e) stems from technical considerations. However, decisions that ensure the complete safety against all possible realizations of the random quantities of the system are frequently unavailable or prohibitively expensive. One is therefore bounded to work with *confidence levels* of operation, i.e., the decisions have to respect a given confidence level, often imposed by regulation or quality standards. Denoting by  $\alpha \in [0, 1]$  the confidence level, the decision making problem may be formulated as

$$\min_{\mathbf{X}, \mathbf{U}} J(\mathbf{X}, \mathbf{U}), \quad (4.4a)$$

subject to (4.3b)–(4.3d), (4.3f) and

$$\mathbb{P}(g(\mathbf{X}) \leq 0) \geq \alpha. \quad (4.4b)$$

Here,  $g$  is a real vector valued function  $g = (g_1, \dots, g_{m_g})$ , representing thus  $m_g$  constraints. The stochastic constraint (4.4b) refers to the *joint* stochastic constraint

$$\mathbb{P}(g_1(\mathbf{X}) \leq 0, \dots, g_{m_g}(\mathbf{X}) \leq 0) \geq \alpha. \quad (4.5)$$

In [29] stochastic constraints are considered individually, i.e. as

$$\mathbb{P}(g_i(\mathbf{X}) \leq 0) \geq \alpha_i, \quad i = 1, \dots, m_g. \quad (4.6)$$

This may indeed represent a valid model choice whenever the random variable present in the individual constraints  $g_i$  are independent of each other. Citing the illustration in [71], this might be the case if the a constraint  $g_i$  models the technical requirements of non-connected water systems that operate under independent climatic conditions. In this case, if  $\alpha_i$  are chosen such that

$$\sum_{i=1}^m 1 - \alpha_i \leq 1 - \alpha, \quad (4.7)$$

then any control vector yielding a trajectory satisfying (4.6) also satisfies (4.5).

Reference [60] deals with joint probability constraints considering the random variables independent. The study of joint stochastic constraints (4.5) without the assumption of independence of the random variables withing each one of them is studied in [69, 70].

As already mentioned, constraints (4.3b)-(4.3d) represent the physical behavior of the system. As such, we require that the solution vector of the optimization problem (4.4) satisfies these constraints almost surely. In the other hand, constraints (4.3e) usually models technical requirements, needed for the nominal operation of the system. Therefore, we are interested in studying the properties of the optimization problem when replacing the almost sure-type technical constraints with probability constraints. In this case, the joint stochastic constraints (4.4b) translate the fact that the state should operate within a specified nominal or safe mode of operation at a given confidence level.

We are interested in a special case of joint stochastic constraints (4.5). In this particular case, one has  $m_g = K$  constraints that impose a certain threshold on some of the state variables *at all times*. This requirement can be formulated as

$$\mathbb{P}(g_{t_0}(X_{t_0}) \leq 0, \dots, g_T(X_T) \leq 0) \geq \alpha. \quad (4.8)$$

The presentation is made clear if the joint constraint is stated in a more convenient form, namely, of an expectation over the final state. We start by rewriting condition (4.8) by adding an additional state variable. Consider the vector  $\mathbf{B} = (B_{t_0}, \dots, B_T)$  where  $B_t \in \{0, 1\}$  such that

$$\begin{aligned} B_{t_0} &= 1, \\ B_{t+1} &= B_t \mathbf{1}_{\mathbb{X}_{t+1}^{\text{ad}}}(X_{t+1}), \end{aligned}$$

where  $\mathbb{X}_t^{\text{ad}} = \{x \in \mathbb{X} \mid g_t(x) \leq 0\}$ . Observe that the binary vector  $\mathbf{B}$  keeps track of the violation of the constraints  $g_t(X_t) \leq 0$ . If at any time a violation occurs, the value of  $B_t$  drops from 1 to 0 and keeps at 0 for all subsequent times. Therefore, (4.8) can be rewritten as a constraint only on the final state.

$$\mathbb{P}(B_T = 1) \geq \alpha. \quad (4.9)$$

Equation (4.9) is in turn rewritten as

$$\mathbb{E}[\theta(B_T)] \geq \alpha, \quad (4.10)$$

where  $\theta(B_T) = \mathbf{1}_{B^\#}(B_T)$  with  $B^\# = \{b \in \{0, 1\} \mid b = 1\}$ . We stress that writing chance constraints as constraints in expectation as in Equation (4.10) should be seen as a notational convenience as there are complexities introduced in the problem by considering an indicator function, for instance, related to convexity and differentiability.

For convenience, we consider that the state variable  $B_t$  is incorporate in the state vector  $X_t$  as a new component.

The cost functional  $J$  has an additive-kind cost structure, which can be seen as running costs  $L_t : \mathbb{X} \times \mathbb{U} \times \Xi \rightarrow \mathbb{R}^+$  and a final cost  $\varphi : \mathbb{X} \rightarrow \mathbb{R}^+$ , namely,

$$J(\mathbf{X}, \mathbf{U}) = \mathbb{E} \left[ \sum_{t=t_0}^{T-1} L_t(X_t, U_t, \xi_{t+1}) + \varphi(X_T) \right]. \quad (4.11)$$

The equality constraints  $h$  have the form

$$h(\mathbf{X}, \mathbf{U}, \xi) \equiv \begin{cases} X_{t_0} - x \\ X_{t+1} - f_t(X_t, U_t, \xi_{t+1}), \end{cases}$$

for  $t = t_0 + 1, \dots, T - 1$  where  $x \in \mathbb{X}$  is a given initial condition. The controlled vector fields  $f_t : \mathbb{X} \times \mathbb{U} \times \Xi \rightarrow \mathbb{X}$  model the evolution of the state process.

Finally, we shall assume that the random variables  $(\xi_{t_0}, \dots, \xi_T)$  are independent. This particular structure is not a restrictive assumption since all dynamical behavior between  $\xi_t$  and past realizations can be transferred to the state vector at the cost of increasing the dimension of the state vector.

Summarizing, the formulation of the optimization problem of interest is

$$\min_{\mathbf{X}, \mathbf{U}} \mathbb{E} \left[ \sum_{t=t_0}^{T-1} L_t(X_t, U_t, \xi_{t+1}) + \varphi(X_T) \right] \quad (4.12a)$$

subject to state and control constraints

$$\mathbf{X} \in \mathcal{X}, \quad (4.12b)$$

$$\mathbf{U} \in \mathcal{U}(\mathbf{X}), \quad (4.12c)$$

dynamic constraints

$$X_{t_0} = x, \quad (4.12d)$$

$$X_{t+1} = f_t(X_t, U_t, \xi_{t+1}), \quad (4.12e)$$

information constraints

$$U_t \preceq \mathcal{F}_t \quad (4.12f)$$

and a stochastic target constraint

$$\mathbb{E}[\theta(X_T)] \geq \alpha. \quad (4.12g)$$

We digress here about the choice of the expected value as an optimization criterion. This choice is justifiable if the controller is risk-neutral and has to act upon the system in several occasions. As such, the law of Large Numbers ensures that, after many realizations of optimal processes, the average cost approaches the optimal expected value. Of course, if no safeguard is made to limit potentially large losses due the variation of the performance of the controller, the decision maker may face a bankruptcy issue. Optimization problems formulated to account for the risk of large losses are referred to *risk-averse* problems. Risk-averse modeling is also justifiable when the controller must

act upon process  $\mathbf{X}$  only a few times, such that the optimization of the average performance is less meaningful. Because in our application, limiting the size of the loss upon a failure is relatively unimportant, the main concern being avoiding failing at all, we choose to optimize an expected value of a chance constrained process rather than using risk-averse models (for details in risk-averse optimization, see [76, 77, 74]).

The choice of introducing probability constraints in this control problem is due the binary character of the failure considered, that is,  $g(x) \leq 0$  (succes) or  $g(x) > 0$  (failure). In order words, the formulation of the optimization problem under probability constraints is recommended whenever one wishes to limit the frequency of *failure events* in successive utilization of the optimal policy. Any quantitative information on the size of the loss in a failure event is not taken into account. This feature characterizes the chance constraints as a *qualitative risk measure* (to follow the nomenclature in [4]). It is indeed well desirable, in some applications, to be interested in how much loss is incurred whenever a failure occurs. In this case, is more adequate to consider *quantitative risk measures* such as the conditional value-at-risk (CVAR) (cf. [75, 78] for an introduction on optimization using risk measures).

In the next section we present an algorithm based on the dynamic programming principle applied to an adequate version of Problem (4.12).

### 4.3 Dynamic Programming

Chance constraint problem (4.12) is equivalent to an optimization problem without probability constraints but with an augmented state and control vector.

Consider the optimization problem

$$\min_{\mathbf{X}, \mathbf{Z}, \mathbf{U}, \mathbf{V}} \mathbb{E} \left[ \sum_{t=t_0}^{T-1} L_t(X_t, U_t, \xi_{t+1}) + \varphi(X_T) \right], \quad (4.13a)$$

subject to state and control constraints

$$\mathbf{X} \in \mathcal{X}, \quad (4.13b)$$

$$\mathbf{U} \in \mathcal{U}(\mathbf{X}), \quad (4.13c)$$

$$\mathbf{Z} \in \mathcal{Z}, \quad (4.13d)$$

$$\mathbf{V} \in \mathcal{V}, \quad (4.13e)$$

dynamic constraints

$$X_{t_0} = x, \quad (4.13f)$$

$$X_{t+1} = f_t(X_t, U_t, \xi_{t+1}), \quad (4.13g)$$

$$Z_{t_0} = 0, \quad (4.13h)$$

$$Z_{t+1} = Z_t + V_t, \quad (4.13i)$$

information constraints

$$U_t \preceq \mathcal{F}_t, \quad (4.13j)$$

$$V_t \preceq \mathcal{F}_{t+1}, \quad (4.13k)$$

an almost-sure constraint in the final state

$$\theta(X_T) - Z_T \geq \alpha, \quad (4.13l)$$

and constraints in expectation

$$\mathbb{E}[V_t | \mathcal{F}_t] = 0, \quad (4.13m)$$

where  $\mathcal{Z} = \mathbb{Z}^K$ ,  $\mathcal{V} = \mathbb{V}^{K-1}$  with  $\mathbb{Z}, \mathbb{V} \subset \mathbb{R}$ .

The next theorem is found in [41, Theorem 5.5]. It is an adaptation of the techniques in [19] to the discrete-time case.

**Theorem 4.1.** *Problems (4.12) and (4.13) are equivalent.*

*Proof.* The equivalence is in the sense that both problems have the same admissible space. Therefore, since the criterion is the same, they amount to the same problem.

We start by showing the equivalence (4.12)  $\Rightarrow$  (4.13). Suppose  $\mathbf{X}, \mathbf{U}$  are admissible processes of Problem (4.12), i.e., they satisfy constraints (4.12b)–(4.12g). Define

$$\begin{aligned} Z_{t+1} &= Z_t + V_t, & Z_{t_0} &= 0, \\ V_t &= \mathbb{E}[\theta(X_T) | \mathcal{F}_{t+1}] - \mathbb{E}[\theta(X_T) | \mathcal{F}_t]. \end{aligned}$$

Following this definition, one obtains  $Z_t = \mathbb{E}[\theta(X_T) | \mathcal{F}_t] - \mathbb{E}[\theta(X_T) | \mathcal{F}_{t_0}]$  and in particular

$$Z_T = \mathbb{E}[\theta(X_T) | \mathcal{F}_T] - \mathbb{E}[\theta(X_T) | \mathcal{F}_{t_0}].$$

Since  $\theta(X_T)$  is measurable with respect to  $\mathcal{F}_T$ ,  $\mathcal{F}_{t_0}$  is the trivial  $\sigma$ -algebra  $\{0, \Xi\}$  and the process  $\mathbf{X}$  satisfies (4.12g), one obtains

$$\theta(X_T) - Z_T = \mathbb{E}[\theta(X_T)] \geq \alpha.$$

Moreover, notice that  $V_t \preceq \mathcal{F}_{t+1}$  by construction.

Now we show (4.13)  $\Rightarrow$  (4.12). Let processes  $\mathbf{Z}, \mathbf{V}, \mathbf{X}, \mathbf{U}$  be admissible to Problem (4.13). From (4.13h)–(4.13i), one obtains  $Z_{t+1} = \sum_{i=t_0}^t V_i$ , yielding in particular

$$\mathbb{E}[Z_T] = \sum_{i=t_0}^{T-1} \mathbb{E}[V_i].$$

Writing  $\mathbb{E}[V_i] = \mathbb{E}[\mathbb{E}[V_i | \mathcal{F}_i]]$ , because of condition (4.13m),  $\mathbb{E}[Z_T] = 0$ . The almost-sure final constraint (4.13l) then yields

$$\mathbb{E}[\theta(X_T)] - \mathbb{E}[Z_T] \geq \alpha \Rightarrow \mathbb{E}[\theta(X_T)] \geq \alpha.$$

□

At the expense of introducing an extra state process  $\mathbf{Z}$  and an extra control process  $\mathbf{V}$ , as well as additional measurability conditions (4.13k) and conditional expectation constraints (4.13m) over  $\mathbf{V}$ , one effectively replaces an expected value-type constraint (4.12g) with an a.s.-type constraint (4.13l). The main advantage of formulation (4.13) is that now one is able to derive a dynamic programming principle for a problem with a.s.-constraints. This allows for the possibility of using a dynamic programming algorithm to synthesize optimal feedbacks.

Another advantage of such a formulation is that the optimal feedback obtained is dynamically consistent (for a very clear exposition on dynamic consistency, see [28]).

The next step in this section is to derive the dynamic programming equations for problem (4.13).

**Definition 4.1.** The value function is defined as the solution value of problem (4.13),

$$v_{t_0}(x, z) = \min_{U, V} \mathbb{E} \left[ \sum_{t=t_0}^{T-1} L_t(X_t, U_t, \xi_{t+1}) + \varphi(X_T) \right]. \quad (4.14a)$$

under constraints (4.13b)-(4.13m) and initial conditions

$$X_{t_0} = x, \quad Z_{t_0} = 0 \quad (4.14b)$$

We begin considering the problem starting from  $t_0 = T$ . By the definition (4.14), the value function is

$$v_T(x, z) = \varphi(x) + \chi_{C^{\text{adm}}}(x, z), \quad (4.15)$$

where  $\chi_C$  is the characteristic function of set  $C$  and we define  $C^{\text{adm}} \triangleq \{(x, z) \mid \theta(x) - z \geq \alpha\}$ . Now, posing  $t_0 = T - 1$  and abridging the notation with

$$R_T(x, z, U, V, \xi) \triangleq L_{T-1}(x, U, \xi) + v_T(f_{T-1}(x, U, \xi), z + V),$$

the optimal value is given by

$$\begin{aligned} v_{T-1}(x, z) &= \min_{\substack{U \preceq \mathcal{F}_{T-1} \\ V \preceq \mathcal{F}_T \\ \mathbb{E}[V | \mathcal{F}_{T-1}] = 0}} \mathbb{E} \left[ L_{T-1}(x, U, \xi_T) + \varphi(f_{T-1}(x, U, \xi_T)) + \chi_{C^{\text{adm}}}(f_{T-1}(x, U, \xi_T), z + V) \right] \\ &= \min_{\substack{U \preceq \mathcal{F}_{T-1} \\ V \preceq \mathcal{F}_T \\ \mathbb{E}[V | \mathcal{F}_{T-1}] = 0}} \mathbb{E} \left[ R_T(x, z, U, V, \xi_T) \right] \\ &= \min_{U \preceq \mathcal{F}_{T-1}} \min_{\substack{V \preceq \mathcal{F}_T \\ \mathbb{E}[V | \mathcal{F}_{T-1}] = 0}} \mathbb{E} \left[ R_T(x, z, U, V, \xi_T) \right]. \end{aligned} \quad (4.16)$$

Here, in the absence of the constraint  $\mathbb{E}[V | \mathcal{F}_{T-1}] = 0$  on the minimization of  $V$ , one is able to interchange the operator of minimization with respect to  $V$  with the expectation operator thanks to the measurability properties of  $V$ . This operator interchange is studied in [72, Theorem 14.60]. Reference [72] uses a continuous-time setting and considers  $\Xi$  to be a continuous set. We do not address the complexities arising in a continuous-time model and we consider the much simpler case where  $\Xi$  is a finite-valued set with cardinality of  $N$ . When the constraint  $\mathbb{E}[V | \mathcal{F}_{T-1}] = 0$  is taken into account, as in the inner minimization problem in (4.16), a question concerning the measurability of the optimal feedback arises. More precisely, considering the optimization problem

$$V^* \in \arg \min_{\substack{V \preceq \mathcal{F}_T \\ \mathbb{E}[V | \mathcal{F}_{T-1}] = 0}} \mathbb{E} \left[ R_T(x, z, U, V, \xi_T) \right], \quad (4.17)$$

the question is whether the optimal feedback  $V^*$  is  $\mathcal{F}_T$ -measurable. In [41, Theorem 5.6], it is shown that, indeed,  $V^*$  is  $\mathcal{F}_T$ -measurable.

We proceed now to detail the dynamic programming principle for problem (4.12). Function  $R_T$  is independent of  $\xi_{T-1}$ . This allows us to write the optimization of the expected value in a conditional fashion. Thus, consider the following optimization problem:

$$\min_{\substack{V \preceq \mathcal{F}_T \\ \mathbb{E}[V | \mathcal{F}_{T-1}] = 0}} \mathbb{E} \left[ R_T(x, z, U, V, \xi_T) \right], \quad (4.18)$$

Problem (4.18) can be decomposed in  $N$  independent subproblems

$$\min_{\substack{V \preceq \mathcal{F}_T \\ \mathbb{E}[V | \xi_{T-1} = \omega_i] = 0}} \mathbb{E} \left[ R_T(x, z, U, V, \xi_T) \right], \quad (4.19)$$

where  $\omega_i \in \Xi$ ,  $i = 1, \dots, N$  are the possible values of random variables  $\xi$ .



**Remark 4.1.** Problems (4.19) are independent. One can therefore solve them in parallel.

More in detail, since  $V \preceq \mathcal{F}_T$ , one can write  $V$  under the form

$$V = \sum_{i=1}^N V^i \mathbf{1}_{\{\xi_{T-1}=\omega_i\}}, \quad (4.20)$$

where each  $V^i$  is  $\mathcal{F}_T$ -measurable. Proceeding as such, problem (4.19) can be rewritten as

$$\min_{\substack{V^i \preceq \mathcal{F}_T \\ \mathbb{E}[V^i]=0}} \mathbb{E}[R_T(x, z, U, V^i, \xi_T)]. \quad (4.21)$$

Notice that the originally conditional expectation on  $V$  becomes a unconditional expectation constraint on each  $V^i$ . Since  $\xi$  is discrete-valued, one can take further profit of the measurability properties of  $V$  and write

$$V^i = \sum_{j=1}^N V^{ij} \mathbf{1}_{\{\xi_T=\omega_j\}}, \quad (4.22)$$

with each  $V^{ij}$  is a constant. This allows the  $N$  independent problems (4.19) to be rewritten as

$$G_T^i(x, z, U) = \min_{V^{i1}, \dots, V^{iN}} \sum_{j=1}^N R_T(x, z, U, V^{ij}, \omega_j) \mathbb{P}(\omega_j) \quad (4.23a)$$

under the coupling constraint

$$\sum_{j=1}^N V^{ij} \mathbb{P}(\omega_j) = 0. \quad (4.23b)$$

The value function is then

$$v_{T-1}(x, z) = \min_U \sum_{i=1}^N G_T^i(x, z, U) \mathbb{P}(\omega_i). \quad (4.24)$$

Proceeding backwards, for  $t_0 = T - 2, \dots, 0$ , one can obtain expressions that allow to compute recursively the value function defined in (4.14). In a similar way, the optimal feedback can be synthesized. The overall procedure is described in the next proposition:

**Proposition 4.1.** For all  $(x, z) \in \mathcal{X} \times \mathcal{Z}$ , the value function can be computed by the relations:

$$v_T(x, z) = \varphi(x) + \chi_{C^{adm}}(x, z), \quad (4.25)$$

$$v_t(x, z) = \min_{\substack{U_t, V_t \\ \mathbb{E}[V_t|\xi_t]=0}} \mathbb{E}[R_{t+1}(x, z, U_t, V_t, \xi_{t+1})], \quad (4.26)$$

for  $t = T - 1, \dots, 0$ , where  $C^{adm} \triangleq \{(x, z) \mid \theta(x) - z \geq \alpha\}$  and

$$R_{t+1}(x, z, U_t, V_t, \xi_{t+1}) = L_t(x, U_t, \xi_{t+1}) + v_{t+1}(X_{t+1}, Z_{t+1}). \quad (4.27)$$

Moreover, if process  $\xi = (\xi_1, \dots, \xi_T)$ , where the  $\xi_t$  are independent, are discrete random variables with  $N$  possible values  $\omega_1, \dots, \omega_N$ , expression (4.26) is given by

$$v_t(x, z) = \min_{U_t} \sum_{i=1}^N G_{t+1}^i(x, z, U_t) \mathbb{P}(\omega_i), \quad (4.28)$$

where each  $G^i$  is the solution of the optimization problem

$$G_{t+1}^i(x, z, U_t) = \min_{V^{i1}, \dots, V^{iN}} \sum_{j=1}^N R_{t+1}(x, z, U_t, V_t^{ij}, \omega_j) \mathbb{P}(\omega_j) \quad (4.29a)$$

with  $V_t^{ij}$  satisfying

$$\sum_{j=1}^N V_t^{ij} \mathbb{P}(\omega_j) = 0. \quad (4.29b)$$

Controls  $V_t$  are given by

$$V_t = \sum_{i,j=1}^N V^{ij} \mathbf{1}_{\{\xi_t=\omega_i, \xi_{t+1}=\omega_j\}}. \quad (4.30)$$

The optimal feedbacks are denoted by  $U_0^*, \dots, U_{T-1}^*$  and  $V_0^*, \dots, V_{T-1}^*$  and satisfy

$$\begin{aligned} U_t^*(x, z) &\in \arg \min_u \sum_{i=1}^N G_{t+1}^i(x, z, u) \mathbb{P}(\omega_i), \\ V_t^*(x, z, u, \xi_t, \xi_{t+1}) &= \sum_{i,j=1}^N V_t^{*ij}(x, z, u) \mathbf{1}_{\{\xi_t=\omega_i, \xi_{t+1}=\omega_j\}}. \end{aligned}$$

where the  $V_t^{*ij}(x, z, u)$  solve (4.29), interpolating the value function whenever necessary.

Then, given some initial conditions  $x$  and a process  $\xi$ , the optimal state trajectories  $X_0^*, \dots, X_T^*$  and  $Z_0^*, \dots, Z_T^*$  are given by

$$X_0^* = x, \quad (4.31a)$$

$$Z_0^* = 0, \quad (4.31b)$$

$$X_{t+1}^* = f_t(X_t^*, U_t^*(X_t^*, Z_t^*), \xi_{t+1}), \quad (4.31c)$$

$$Z_{t+1}^* = Z_t^* + V_t^*(X_t^*, Z_t^*, U_t^*(X_t^*, Z_t^*), \xi_t, \xi_{t+1}). \quad (4.31d)$$

**Remark 4.2.** The dynamic programming algorithm computes the projection of the value function over points of a discrete grid of the state space. It is well known that the way the state space is discretized matters on the performance of the algorithm and the a priori quality of the solution. In the augmented problem (4.13), although the physical characteristics of the model may provide some insight into how to create a grid of  $\mathcal{X}$ , a similar insight into the discretization of  $\mathcal{Z}$  does not seem straightforward. The additional control process  $\mathbf{V}$  may be interpreted as the variation of the perception of the expectation constraint (4.12g) due the additional information generated by the unfolding of process  $\xi$ . The state process  $\mathbf{Z}$  keeps track of the cumulated level of that perception. Indeed, this interpretation stems from construction

$$Z_{t+1} = Z_t + V_t \quad (4.32)$$

$$V_t = \mathbb{E}[\theta(X_T) \mid \mathcal{F}_{t+1}] - \mathbb{E}[\theta(X_T) \mid \mathcal{F}_t] \quad (4.33)$$

used in the proof of theorem 4.1. Notice that, since  $\theta : \mathbb{X} \rightarrow \{0, 1\}$ , from Equation (4.33), controls  $V_t \in [-1, 1]$ . Moreover, from (4.32) and the particular structure of (4.33), the cumulated perception of the constraint  $Z_t$  also lies in the interval  $[-1, 1]$ .

## 4.4 Chance-constrained Power Management

This section is devoted to present the vehicle model that represents the application of interest. The ultimate goal is to synthesize an optimal controller dictating the power request repartition among an internal combustion engine (ICE) and a high-voltage battery. The optimality is seen as the total economic cost of a car trip, which amounts to the least possible use of the ICE. The random variable present in the model is a random perturbation on the total remaining charge of the battery.

The vehicle in consideration is a particular class of hybrid vehicles called *range extender electric vehicles* (REEV), thoroughly detailed in Chapters 1,2.

We recall the main points of this application. Given a route (for instance, proposed by a navigation system), one faces three possible scenarios: first, the energy on the battery is enough to allow the vehicle to reach its destination ; the second scenario is when the *total* energy available (energy on the battery plus energy on the range extender) is not enough to allow the vehicle to reach its destination ; finally, the last scenario is when the battery does not have enough charge to reach the destination and the utilization of the range extender (RE) is absolutely required to allow the vehicle to reach the destination<sup>1</sup>.

A *failure* is the event of running out of energy before attaining the final destination. In the case of the range extender electric vehicle, a failure event occurs whenever the battery charge drop below a certain given level. Mathematically, this question is translated into a discrete-time stochastic optimal control problem under chance constraints.

In any case, the model imposes the requirement of not running out of energy in the battery. Indeed, the REEV controller fails whenever the SOC drops to zero, even if there is some fuel left in the range extender.

In a first time, we shall introduce a somewhat simplified model of the vehicle. The state is composed of  $x$ , the SOC of the battery. We denote by  $r$ , the fuel left in the RE reservoir. Following Remark 2.3, the fuel available in the RE reservoir is not considered a state variable. The control is denoted  $u$  and represents the power produced by the RE. We shall consider that the SOC consumption may experiment a random impact, denoted by  $\xi$ . In a realistic application, this random variable may represent speed fluctuations around a recommended cruise speed given by the navigation system of the vehicle. If a stochastic model for the vehicle speed is available,  $\xi$  may be interpreted as a forecast error. Those kinds of deviations from a predicted speed impact the predicted energy consumption and are a major factor impacting several applications on the automotive industry.

The SOC and fuel dynamics is given by

$$X_{t+1} = x_t - a_x + u_t + \xi_{t+1}, \quad (4.34)$$

$$r_{t+1} = r_t - a_r u_t, \quad (4.35)$$

with given  $x_0, r_0$ . We assume a unit system in which all the constants needed to adjust the units accordingly are 1. In this simple model, the power produced by the RE causes the fuel level to decrease at a constant rate  $a_r > 0$ . In the other hand, producing power can reduce the rate of depletion of the battery or indeed recharge the battery. The constant  $a_x > 0$  represents an “expected” rate of depletion of the battery. The evolution of the SOC is affected by the random quantity  $\xi$  which may increase or decrease the depletion rate. We assume that capacities of the battery and reservoir are normalized, and therefore,  $x_t, r_t \in [0, 1]$  at all times. Also, the power control is bounded above by  $\bar{u}$  and thus,  $u_t \in [0, \bar{u}]$ .

The objective of a power management strategy is to minimize the fuel usage on a particular car trip. Denote the horizon by  $T$ , representing the total length of a given route. The criterion to optimize is given by

$$J(\mathbf{X}, \mathbf{U}) = \mathbb{E}[(r_0 - r_T)]. \quad (4.36)$$

Following the expression of the fuel dynamics (4.35), one can readily rewrite the criterion to be minimized as

$$J(\mathbf{U}) = \mathbb{E} \left[ \sum_{t=0}^{T-1} u_t \right], \quad (4.37)$$

---

<sup>1</sup>We observe that, given a total initial energy and a route, the identification of the corresponding scenario is a delicate matter.

since  $r_0 - r_T = a_r \sum_{t=0}^{T-1} u_t$ .

Now we introduce the chance constraint. The event the application wants to avoid is the complete discharge of the battery. Since the RE cannot drive without power from the battery, the failure happens whenever the SOC reaches  $x = 0$  before completing the trip. Alternatively, a success refers to whenever the vehicle reaches its final destination keeping a SOC positive at all times. Thus, The chance constraint is written as

$$\mathbb{E}[X_0 \geq 0, \dots, X_T \geq 0] \geq \alpha, \quad (4.38)$$

where  $\alpha$  is a specified confidence level.

Vis-à-vis the application, the interesting case happens when the ratio of initial SOC to the total drive distance is tight so that a “modest” operation of the RE may cause a failure. Another interesting setting is when the random variable may have extreme realizations as to cause an increase in the awareness of the control policy. If the probability of these extreme events is small, the controller is overrating the failure possibility and routinely applying an over-conservative feedback, resulting in a unnecessary operation cost of the RE.

The instance used for the numerical simulations is

- $a_x = 0.07$ ,  $a_r = 1.3$ ,  $x_0 = 0.50$ ,
- $T = 10$ ,  $\bar{u} = 0.07$ .

The random variable presents the following distribution:

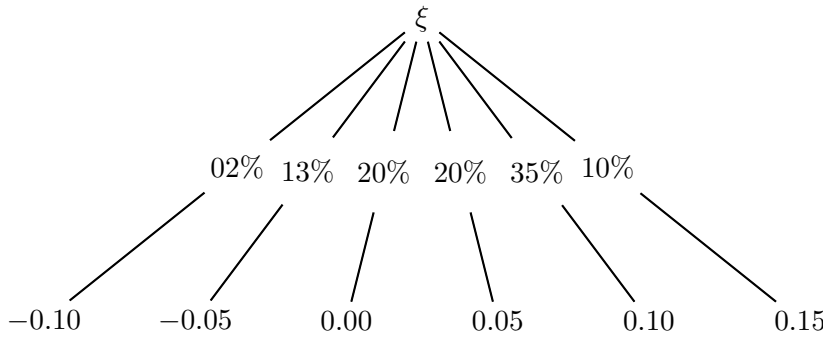


Figure 4.1 – Distribution probability for random perturbation of the battery charge consumption.

Two confidence levels are considered in the simulation,  $\alpha = 0.90, 0.99$ . The value function is computed using the algorithm presented in Proposition 4.1 and  $N = 1000$  feedbacks are synthesized. Figure 4.2 shows the cost of the policies and the state trajectories for each value of  $\alpha$ . Table 4.1 summarizes the results obtained in the simulations. The table presents, for each value of confidence level  $\alpha$  the average cost of policies, the standard deviation as a percentage of the average cost and the obtained success rate. The success rate obtained using the computed policies are indeed close to the required confidence level. For the most stringent confidence level ( $\alpha = 0.99$ ) the average cost is higher, representing an increase of 6% with respect to a confidence level of 90%. Also, following a more strict confidence level, the cost distribution is tighter, presenting a reduction of over 7% in the deviation with respect to the average cost. This behavior reflects the fact the less strict confidence level generates policies that are more lax, making room for more extreme cost savings.

The next step towards a real world application is the study of the chance constrained problem in richer models. In particular, the random variable must be accurately modeled if the chance constrained policies are expected to perform well in simulations using real vehicle data.

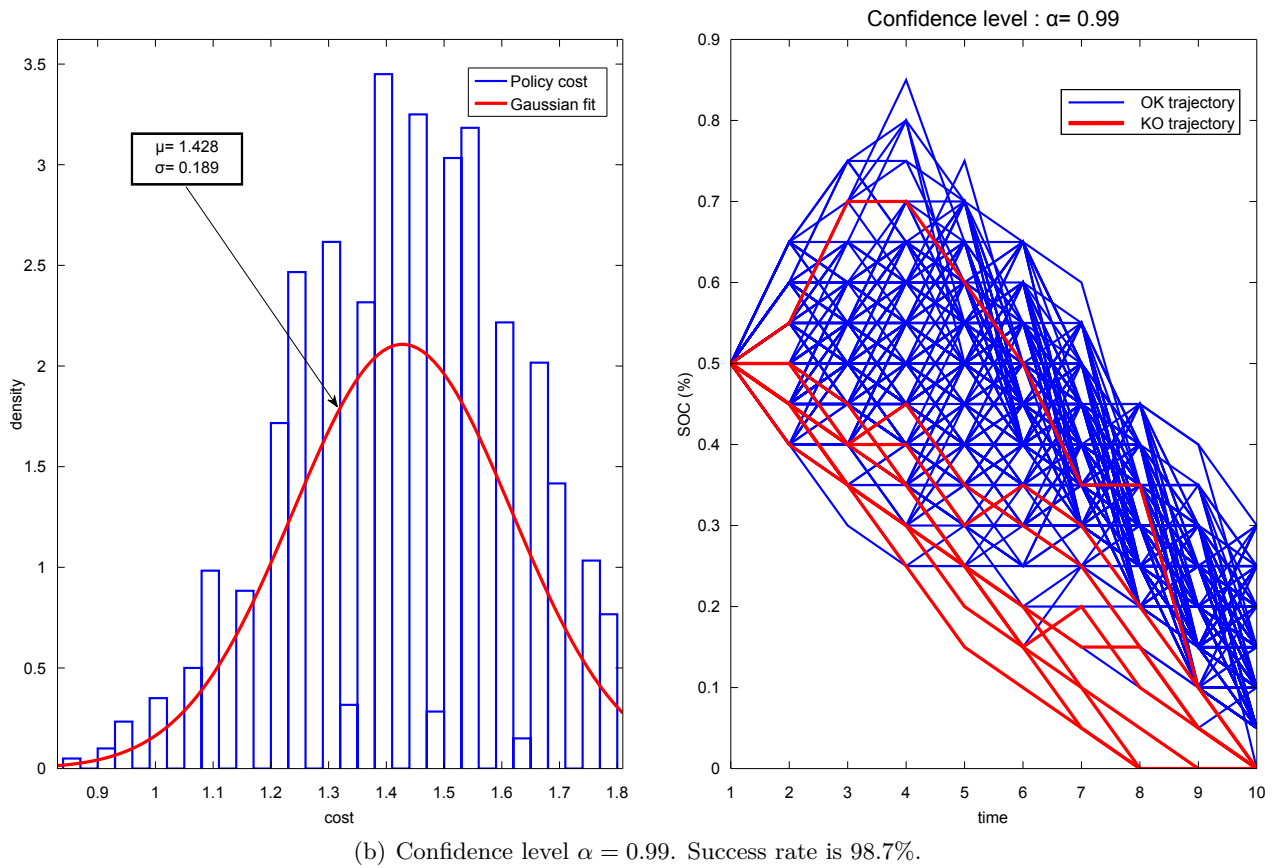
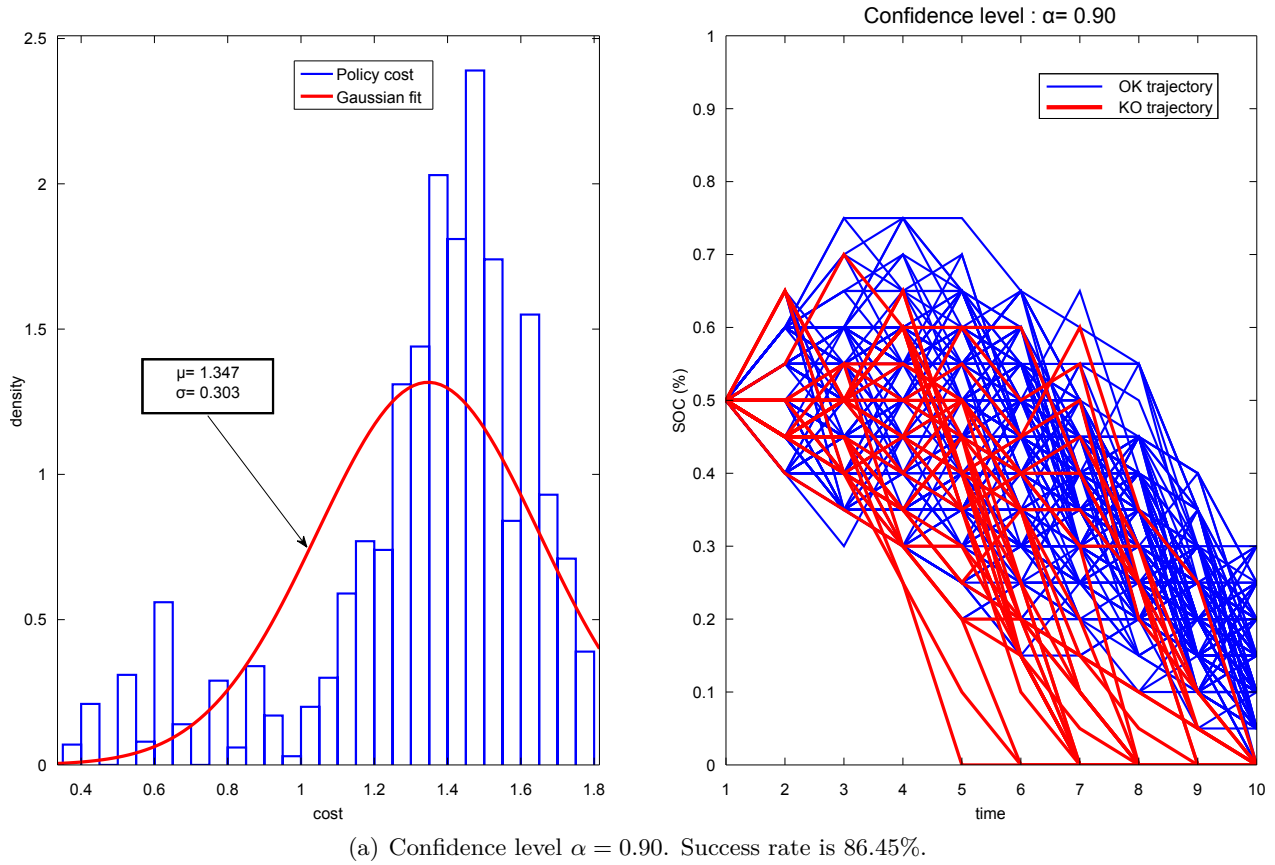


Figure 4.2 – Cost distributions and SOC trajectories for  $N = 1000$  policies. Trajectories in blue are successful while trajectories in red are fails.

confidence level	average cost	deviation	success rate
90%	1.347	21.09 %	86.45 %
99%	1.428	13.24 %	98.70 %

Table 4.1 – *Summary of simulations results using chance constrained power management strategies.*

## 4.5 Discussion

As mentioned, the introduction of chance constraints in the optimal power management problem adds flexibility to the optimal control policy. In scenarios where success cannot be fully assured, the control policy is still able to be optimal, in the sense of the financial cost, while at the same time limiting the frequency of failures to satisfy a certain safety level.

In the optimal power management case, the chance constraints must be verified at each optimization stage. The section proposes a transformation that allow the chance constraint to bear upon the final state only.

This section introduces an equivalent formulation of the stochastic optimal control problem under chance constraints in the final state which enables the problem to be solved via an dynamic programming algorithm. Indeed, by adding a new state variable and a new control variable with adequate measurability properties, a dynamic programming principle is derived which can be used as a base for an algorithm to synthesize the optimal feedback control. The new state variable can be interpreted as the perception of how good the feedback is in satisfying the chance constraints viewed from the initial time. The new control variable can be viewed as adjusting this perception as the system evolves in time.

A dynamic programming principle is derived for this equivalent formulation. The dynamic programming principle in Proposition 4.1 is detailed and presented in such a way that it is readily implementable as a DP algorithm for numerically computing optimal feedbacks.

The simulation results shows that, indeed, lower confidence bounds yield less conservative feedbacks. The consequence is a reduction on the average cost over several policy utilizations associated with the reduction of the success rate of the strategies.

Considering a more realistic application, considering a full hybrid vehicle model and a richer stochastic model for the random variable, tuning the success rate to increase fuel economies and reduce emission levels seems to be a promising idea. This introductory study paves the way for further research in this direction.



# REACHABILITY OF HYBRID SYSTEMS: THEORETICAL STUDY

---

*“If in other sciences we should arrive at certainty without doubt and truth without error, it behooves us to place the foundations of knowledge in mathematics.”*

Roger Bacon

The content of this chapter is an adaptation from the following publication and oral communication:

- Granato, G. and Zidani, H. *Level-set approach for reachability analysis of hybrid systems under lag constraints*, SIAM J. Control and Optimization, submitted, awaiting review.
- Granato G. and Zidani H. *HJB Equations for Non-Autonomous Hybrid Dynamical Systems*, Summer School and Workshop of the Initial Training Network for Sensitivity Analysis and Deterministic Optimal Control - SADCO, 2011, London, England, *invited speaker*.

## Contents

---

<b>5.1</b>	<b>Introduction</b>	<b>107</b>
<b>5.2</b>	<b>Motivation and Problem Settings</b>	<b>109</b>
5.2.1	Range Extender Electric Vehicles	109
5.2.2	Hybrid Dynamical System	110
5.2.3	Reachability of Hybrid Dynamical Systems and Optimal Control Problem	113
<b>5.3</b>	<b>Main Results</b>	<b>115</b>
<b>5.4</b>	<b>Numerical Analysis</b>	<b>123</b>
5.4.1	Numerical Scheme and Convergence	123
5.4.2	Numerical Simulations	126

---





## 5.1 Introduction

This chapter deals with the characterization of a reachable set of a hybrid dynamical system with a lag constraint in the switch control. The approach consists in the introduction of an adequate hybrid optimal control problem with lag constraints on the switch control whose value function allows a characterization of the reachable set.

This work is motivated by an application in the automobile industry, namely, calculating the autonomy of a class of electric vehicles (EVs), the range extender electric vehicle (REEV) class, which possesses two distinct sources of energy from which we can use to tract the vehicle. In this setting, the study aims at finding the control sequence of the two energy sources that allows the vehicle to reach the furthest possible point of a given route. The REEV is modeled as a hybrid dynamical system in which the state vector represents the energy capacities of the two different energy sources embedded in the vehicle.

The term hybrid system refers to a general framework that can be used to model a large class of systems. Broadly speaking, they arise whenever a collection continuous- and discrete-time dynamics are put together in a single model. In that sense, the discrete dynamics may dictate switching between the continuous dynamics, jumps in the system trajectory or both. Moreover, they can contain specificities, as for instance, autonomous jumps and/or switches, time delay between discrete decisions, switching/jumping costs. This work considers a particular class of hybrid system where only switching between continuous dynamics are operated by the discrete logic, with no jumps in the trajectory, and there are no switching costs. In addition, switch decisions are constrained to be separated in time by a non-zero interval, fact which is referred to as switching lag.

Before referring to the reachability problem in the hybrid setting, the main ideas are introduced in the non-hybrid framework. Given a time  $t > 0$ , a closed target set  $X_0$  and a closed admissible set  $K$ , considering a controlled dynamical system

$$\dot{y}(\tau) = f(\tau, y(\tau), u(\tau)), \quad \text{a.e. } \tau \in [0, t], \quad (5.1)$$

where  $f : \mathbb{R}^+ \times \mathbb{R}^d \times U \rightarrow \mathbb{R}^d$  and  $u : \mathbb{R}^+ \rightarrow U$  is a measurable function, the reachable set  $R_{X_0}$  at time  $t$  is defined as the set of all initial states  $x$  for which there exists a trajectory that stays inside  $K$  on  $[0, t]$  and arrives at the target:

$$R_{X_0}(t) := \{x \mid (y, u) \text{ satisfies (5.1), with } y(0) = x \text{ and } y(t) \in X_0 \text{ and } y(s) \in K \text{ on } [0, t]\}.$$

It is a known fact that the reachable set can be characterized by the negative region of the value function of an optimal control problem. For this, following the idea introduced by Osher [64], one can consider the control problem defined by:

$$v(x, t) := \inf\{v_0(y(t)) \mid (y, u) \text{ satisfies (5.1), with } y(0) = x \text{ and } y(s) \in K \text{ on } [0, t]\}, \quad (5.2)$$

where  $v_0$  is a Lipschitz continuous function satisfying  $v_0(x) \leq 0 \iff x \in X_0$  (for instance,  $v_0$  can be the signed distance  $d_{X_0}$  to  $X_0$ ). Under classical assumptions on the vector-field  $f$ , one can prove that the reachable set is given by

$$R_{X_0}(t) = \{x \in K, v(x, t) \leq 0\}.$$

Moreover, when  $K$  is equal to  $\mathbb{R}^d$ , the value function has been shown to be the unique viscosity solution of a Hamilton-Jacobi-Bellman (HJB) equation [8]:

$$\partial_t v + \sup_{u \in U} (-f(s, x, u) \cdot \nabla v) = 0 \quad \text{on } \mathbb{R}^d \times (0, t],$$

with the initial condition  $v(x, 0) = v_0(x)$ .

When the set  $K$  is a subset of  $\mathbb{R}^d$  ( $K \neq \mathbb{R}^d$ ), the characterization of  $v$  by means of a HJB equation becomes a more delicate matter and usually requires some additional controllability properties [27, 17]. However, it was pointed out in [17] that in case of state-constraints, the auxiliary control problem should be introduced as

$$V(x, t) := \inf \left\{ v_0(y(t)) \bigvee \max_{\theta \in [0, t]} g(y(\theta)) \mid (y, u) \text{ satisfies (5.1), with } y(0) = x \right\}, \quad (5.3)$$

where  $g$  is any Lipschitz continuous function satisfying  $g(x) \leq 0 \iff x \in K$  (again,  $g$  can be the signed distance  $d_K$  to  $K$ ). This new control problem involves a supremum cost but does not include any state constraints. The reachability set is still given by

$$R_{X_0}(t) = \{x \in K, V(x, t) \leq 0\},$$

and the value function  $V$  is the unique viscosity solution of a variational inequality

$$\min(\partial_t V + \sup_{u \in U} (-f(s, x, u) \cdot \nabla V), V - g) = 0 \quad \text{on } (0, t] \times \mathbb{R}^d,$$

with the initial condition  $v(x, 0) = \min(v_0(x), g(x))$ , but no controllability assumption is needed.

In this paper, we are interested in the extension of the reachability framework to some class of control problems of Hybrid systems.

Let us recall that a hybrid dynamical system is a collection of controlled continuous-time processes selected through a high-level discrete control logic. A general framework for the (optimal) control hybrid dynamical systems was introduced in [22]. Several papers deal with the optimal control problem of hybrid systems, let us just mention here the papers of [40, 5, 25, 84] where the optimality conditions in the form of Pontryagin's principle are studied and [90, 26, 32] where the HJB approach is analyzed.

A feature of the hybrid system used in our work is a time lag between two consecutive switching decisions. From the mathematical viewpoint this remove the particularities linked to Zeno-like phenomena [89]. Indeed, the collection of state spaces is divided in subsets labeled in three categories according to whether they characterize discrete decisions as optional, required (autonomous) or forbidden. Landing conditions ensure that whatever the region of the state space the state vector "lands" after a switch no other switch is possible by requiring a positive distance (in the Hausdorff sense) between the landing sets and the optional/autonomous switch sets. In the other hand, when allowed to switch freely without any costs, when no time interval is imposed between discrete transitions, a controller with a possibly infinite number of instantaneous switches may become admissible. Switch costs can be introduced in order to rule out this kind of strategy by the controller as it becomes over-expensive to switch to a particular mode using superfluous transitions. However, such costs do not make sense in the level-set approach used in this paper.

Diffusion processes with impulse controls including switch lags are studied in [24], where it is considered the idea of introducing a state variable to keep track of the time since the last discrete control decision. There, in addition, discrete decisions also suffer from a time delay before they can manifest in the continuous-time process. In that case, one has the possibility of scheduling discrete orders whenever the time for a decision to take place may be longer than that of deciding again. Then, the analysis also includes keeping record of the nature of this scheduled orders. This work inspired the idea of a state variable locking possible transitions used here.

To study the reachability sets for our system, we follow the level set approach and adapt the ideas developed in [17] to hybrid systems by proposing a suitable control problem which allows us to handle in a convenient way the state constraints and the decision lag. It is proven that

$$R_{X_0}(s) = \{x \mid \exists (q, p) \in Q \times P, v(x, q, p, s) \leq 0\}.$$

where  $v$  denotes the value function of a hybrid optimal control problem. Thus, through a characterization of  $v$ , one obtains the desired reachable set, defined in the hybrid context,  $x$  is in the (physical) state of the system,  $q$  is the discrete variable and  $p$  is a switch lock variable (defined further below). Here the main difficulty is to characterize the value function associated to the control problem. It turns out that this value function satisfies a quasi-variational HJB inequality system (in the viscosity sense)

$$\min(\partial_s u + \partial_p u + H(s, x, q, \nabla_x u), u - \varphi(x)) = 0 \quad , \quad (x, q, p, s) \in \Omega, \quad (5.4)$$

$$u(x, q, p, s) - (Mu)(x, q, p, s) = 0 \quad , \quad p = 0 \quad (5.5)$$

$$u(x, q, p, 0) = \max(\phi(x), \varphi(x)). \quad (5.6)$$

where  $t > 0$ ,  $d > 0$ ,  $X = \mathbb{R}^d$ ,  $Q = \{0, 1\}$ ,  $P = (0, t]$ ,  $T = (0, t]$ ,  $\Omega = X \times Q \times P \times T$  and where  $\phi, \varphi : X \rightarrow \mathbb{R}$  are target and obstacle indicator functions (defined properly further below). Here,  $M$  is a non-local switch operator that acts whenever the state variable  $p$  touches the boundary  $p = 0$ . Moreover, we give a comparison principle of this system. Usually, the proof of the comparison principle requires some transversality assumptions that do not make sense in the kind of applications we are interested in. However, while the decision lag complicates the structure of the problem it also plays a role in the proof of comparison principle (the same role that the transversality assumed in [90, 33]) acting as a kind of landing condition in the sense explained above. The proof of the comparison principle is close to the one given in [32, 9] and adapts the idea of using “friendly giants”-type functions.

This chapter is organized as follows: firstly, it describes briefly the industrial application motivating the study. (in a slightly more general setting than that required for the application). Then, a hybrid optimal control problem is stated, the reachable set and the value function are defined and a dynamic programming principle for the value function is obtained. The value function is shown to be Lipschitz continuous and also a solution of an system of quasi-variational inequalities (SQVI). It follows with the proof of a comparison principle for the SQVI that ensures uniqueness of its solution and is used to show the convergence of a class of numerical schemes for the computation of the value function. Lastly results of some numerical simulations evaluating the autonomy of a REEV toy model and illustrating the convergence of a discretization scheme are presented.

## 5.2 Motivation and Problem Settings

### 5.2.1 Range Extender Electric Vehicles

This section proposes an algorithm to synthesize controllers for the power management on board hybrid vehicles that allows the vehicle to reach its maximum range along a given route. Although the algorithm is readily applicable to all classes of hybrid vehicles, this work focuses on a specific class, namely, range extender electric vehicles (REEV). A range extender (RE) electric vehicle is an electric vehicle that disposes of an additional source of energy besides the main high voltage (HV) traction battery. The controls available are the RE’s state – on or off – and the power produced in the RE (and delivered into the powertrain). The power delivered into the powertrain is a non-negative piecewise continuous time function. The RE’s state is controlled by a discrete sequence of switching orders decided and executed at discrete times. An important feature of the REEV model is a time interval  $\delta > 0$  imposed between two consecutive decisions times. From the physical viewpoint, this assumption incorporates the fact that frequent switching of the RE is undesirable in order to avoid mechanical wear off and acoustic nuisance for the driver. The model considers that the vehicle’s traction capability is conditioned to the existence of some electric energy in the battery. Since the vehicle must halt whenever there is no charge available in the battery, the objective of finding the vehicle autonomy is summarized into finding the furthest point away from the vehicle geographic starting point where the battery is depleted for the first time.

The REEV is modeled as a discrete-time dynamical system in which the state vector represents the energy capacities of the two different energy sources embedded in the vehicle. In this setting, the study aims at finding the control sequence of the two energy sources that allows the vehicle to reach the furthest possible point of a given route.

### 5.2.2 Hybrid Dynamical System

Hybrid systems have some supervision logic that intervenes punctually between two or more continuous functions. The main elements of the class of hybrid dynamical systems considered in this work are a family of continuous dynamics (vector fields)  $f$  and continuous state spaces  $X$ , indexed by a discrete state  $q$  valued in a discrete set  $Q$ . Each continuous dynamic system  $f_q$  valued in  $X_q$  models a physical process controlled by a continuous control function  $u$ , from which the system is free to switch to another process  $f_{q'}$  using a specified discrete control  $w$  and a discrete dynamics  $g$ .

More precisely, the continuous state variable is denoted  $y$  and it is valued in the state space  $X = \mathbb{R}^d$ . The discrete variable is  $q \in Q = \{0, 1, \dots, d_q\}$ , where  $d_q$  is the number of possible dynamics that can operate the system. (for simplicity of the presentation, through this paper, we consider that  $X_q = X$  for all  $q \in Q$ ). Each of these dynamics models a different mode of operation of the system or a different physical process that the system is undergoing. Moreover, define a compact set  $K \subset X$  as the hybrid system admissible set, i.e., a set inside which the state must remain.

The continuous control is supposed to be a measurable function  $u$  valued in a set that depends on the mode that is currently active  $U(q)$ . The discrete control is a sequence of switching decisions

$$w = \{(w_1, s_1), \dots, (w_i, s_i), (w_{i+1}, s_{i+1}), \dots\}, \quad (5.7)$$

where each  $s_i \in [0, \infty[$  and  $w_i \in W(q) \subset \{0, 1, \dots, d_q\}$ . The sequence of discrete switching decisions  $\{w_i\}_{i \geq 0}$  (designating the new mode of operation) is associated with a sequence of switching times  $\{s_i\}_{i \geq 0}$  where each decision  $w_i$  is exerted at time  $s_i$ . The set of available discrete decisions, at time  $s$ ,  $W(q(s))$ , depends on the discrete state variable and it corresponds to a decision of switching the system to another process  $w_i$ .

The lag condition between switches is included by demanding that two switch orders must be separated by a time interval of  $\delta > 0$ , i.e.,

$$s_{i+1} - s_i \geq \delta.$$

**Remark 5.1.** Regarding the vehicle application, the vehicle's energy state is a two-dimensional vector  $y \in X = \mathbb{R}^2$ , where  $y = (y_1, y_2)$  denotes the state of charge of the battery and the fuel available in the range extender module. Each of these quantities are the image of the remaining energy in the battery and the RE respectively. It is clear here that the state variables have to be constrained to remain in the compact set  $K = [0, 1]^2$ , where the energies quantities are normalized.  $q \in Q = \{0, 1\}$  is the RE state, indicating whether the RE is off ( $q = 0$ ) or on ( $q = 1$ ). The power output is a measurable function  $u(\cdot) \in U(q(\cdot))$  where  $U(\cdot)$  is the admissible control set, compact subset of  $\mathbb{R}$  dependent naturally on the RE state.

In this setting, given a discrete state  $q$ , the continuous control  $u$  steers the continuous system

$$\dot{y}(\tau) = f(\tau, y(\tau), u(\tau), q_i), \text{ for a.e. } \tau \in [0, t] \quad (5.8)$$

where some continuous dynamics  $f(\cdot, \cdot, \cdot, q_i)$  is activated.  $f$  is a family of vector fields indexed by the discrete variable  $q$ . When  $q = q_i$ , the corresponding vector field is active and dictates the evolution

of the continuous state. At some isolated times  $\{s_i\}_{i>0}$ , given by the discrete switching control sequence, the discrete dynamic  $g$  is activated

$$q_{i-1} = g(w_i, q_i) \quad (5.9)$$

and the continuous state follows another vector field  $f(\cdot, \cdot, \cdot, q_{i-1})$ . In the considered system, the discrete decisions only switch the continuous dynamics and introduce no discontinuities on the trajectory. In more general frameworks, we can also include jumps in the continuous state vector that can be used to model an instant change in the value of the state following a discrete decisions [22, 14].

Assume controlled continuous dynamics  $f$  and the discrete dynamic  $g$  satisfy the following:

**(H0)** The continuous control is a measurable function  $u : [0, \infty[ \rightarrow \mathbb{R}^m$  such that

$$u(\tau) \in U(q(\tau)) \text{ for a.e. } \tau \in [0, t].$$

**(H1)** There exists  $L_f > 0$  such that, for all  $s \geq 0$ ,  $y, y' \in X$ ,  $q \in Q$  and  $u \in U(q)$ ,

$$\|f(s, y, u, q) - f(s, y', u, q)\| \leq L_f \|y - y'\|, \quad \|f(s, y, u, q)\| \leq L_f.$$

**(H2)** For all  $q \in Q$ ,  $f(\cdot, \cdot, \cdot, q) : [0, \infty[ \times X \times U \rightarrow X$  is continuous and for all  $s \in [0, \infty[$ ,  $x \in X$ ,  $u \in U$ ,  $f(s, x, u, \cdot) : Q \rightarrow X$  is continuous with respect to the discrete topology.

**(H3)** For all  $s \in [0, t]$ ,  $y \in X$  and  $q \in Q$ ,  $f(s, y, U, q)$  is a convex subset of  $X$ .

**(H4)** There exists  $L_g > 0$  such that, for all  $q \in Q$  and  $w \in W(q)$ ,

$$\|g(w, q)\| \leq L_g$$

**(H5)**  $g(\cdot, \cdot)$  is continuous with respect to the discrete topology.

Assumption (H1) ensures that a trajectory exists and that it is unique. Assumptions (H2)-(H5) are used to prove the Lipschitz continuity of the value function and (H3) is needed in order to observe the compactness of the trajectory space.

Denote  $A$  the space of hybrid controls  $a = (u, w)$ . We precise the class of admissible controls  $\mathcal{A} \subset A$  in the following definition:

**Definition 5.1.** For a fixed  $t \geq 0$  a hybrid control  $a = (u, w) \in \mathcal{A}$  is said to be admissible if the continuous control verifies (H0) and the discrete control sequence  $w = \{w_i, s_i\}_{i>0}$  has increasing decision times

$$s_1 \leq s_2 \leq \dots \leq s_i \leq s_{i+1} \leq \dots \leq t, \quad (5.10)$$

admissible decisions

$$\forall i > 0, w_i \in W(q(s_i)) \subset Q, \quad (5.11)$$

and verifies a decision lag

$$s_{i+1} - s_i \geq \delta, \quad (5.12)$$

where  $\delta > 0$ .

An important consequence in the definition of admissible control is the finiteness of the number of switch orders:

**Proposition 5.1.** Fix  $s \geq 0$ . Let  $a \in A$  be an admissible hybrid control. Then, the discrete control sequence has at most  $N = \lfloor s/\delta \rfloor$  switch decisions.

Fix  $t > 0$ . Given a hybrid control  $a \in A$  with  $N$  switch orders and given  $x \in X$ ,  $q \in Q$ , the hybrid dynamical system is

$$\dot{y}(\tau) = f(\tau, y(\tau), u(\tau), q_i), \quad \tau \in [0, t], \quad y(t) = x \quad (5.13)$$

$$q_{i-1} = g(w_i, q_i), \quad i = 1, \dots, N, \quad q_N = q \quad (5.14)$$

Denote the solutions of (5.13)-(5.14) with final conditions  $x, q$  by  $y_{x,q;t}$  and  $q_{x,q;t}$ . As pointed out, not all discrete control sequences are admissible. Only admissible control sequences engender admissible trajectories. Thus, given  $t > 0$ ,  $x \in X$  and  $q \in Q$ , the admissible trajectory set  $Y_{[0,t]}^{x,q}$  is defined as

$$Y_{[0,t]}^{x,q} = \{y(\cdot) \mid a \in \mathcal{A} \text{ and } y_{x,q;t} \text{ solution of (5.13)-(5.14)}\} \quad (5.15)$$

A consequence of proposition 5.1 and the above definition is the finiteness of the number of discrete decisions in any admissible trajectory. Observe that the admissible trajectories set does not include the discrete trajectory.

The hybrid control admissibility condition formulated as in conditions (5.10)-(5.12) is not well adapted to a dynamic programming principle formulation, needed later on. In order to include the admissibility condition in the optimal control problem in a more suitable form, we introduce a new state variable  $\pi$ . Recall that the decision lag conditions implies that new switch orders are not available up to a time  $\delta$  since the last switch. The new variable is constructed such that at a given time  $\tau \in [0, t]$ , the value of  $\pi(\tau)$  measures the time since the last switch. The idea is to impose constraints on this new state variable and treat them more easily in the dynamic programming principle. Thus, if  $\pi(\tau) < \delta$  all switch decisions are blocked and if, conversely,  $\pi(\tau) \geq \delta$  the system is free to switch. For that reason, this variable can be seen as a switch lock.

Now given  $t > 0$ ,  $\tau \in [0, t]$  and a discrete control  $w = \{w_i, s_i\}_{i>0}$ , the switch lock dynamics is defined by

$$\pi^w(\tau) = \pi(\tau) = \begin{cases} \delta + \tau & \text{if } \tau < s_1 \\ \inf_{s_i \leq \tau} \tau - s_i & \text{if } \tau \geq s_1 \end{cases} \quad (5.16)$$

Indeed, once the discrete control is given, the trajectory  $\pi(\cdot)$  can be determined. Proceeding with the idea of adapting the admissibility condition in order to manipulate it in a dynamic programming principle, we wish to consider  $\pi(t) = p$ , with  $p \in P := (0, t]$ , the final value of the switch lock variable trajectory and impose the lag condition under the form  $\pi(s_i^-) \geq \delta$  for all  $s_i$ , where  $s_i^-$  denotes the limit to the left at the switching times  $s_i$  (notice that  $\pi(s_i^+) = 0$  by construction). Then, since these conditions suffice to define an admissible discrete control set, while optimizing with respect to admissible functions, one needs only look within the set of hybrid controls that engenders a trajectory  $\pi(\cdot)$  with the appropriate structure. In other words, given  $t > 0$ ,  $x \in X$ ,  $q \in Q$  and  $p \in P$ , define a admissible trajectory set  $S_{[0,t]}^{x,q,p}$  as

$$\begin{aligned} S_{[0,t]}^{x,q,p} &= \{y(\cdot) \mid a = (u, \{w_i, s_i\}_{i=1}^N) \in A, \ y_{x,q;t} \text{ solution of (5.13)-(5.14)}, \\ &\quad \pi(\cdot) \text{ solution of (5.16)}, \ \pi(t) = p, \ \pi(s_i^-) \geq \delta, \ i = 1, \dots, N\}. \end{aligned} \quad (5.17)$$

The next lemma states a relation between sets  $Y$  and  $S$ :

**Lemma 5.1.** *Following the above definitions, sets (6.6) and (6.8) satisfy*

$$Y_{[0,t]}^{x,q} = \bigcup_{p \in P} S_{[0,t]}^{x,q,p}$$

*Proof.* The equivalence between  $Y_{[0,t]}^{x,q}$  and  $\bigcup_{p \in P} S_{[0,t]}^{x,q,p}$  is obtained by construction.  $\square$

In the following of the paper, whenever we wish to call attention to the fact that the final conditions of (5.13), (5.14) and (6.7) are fixed, we denote their solutions respectively by  $y_{x,q,p;t}$ ,  $q_{x,q,p;t}$ ,  $\pi_{x,q,p;t}$ .

### 5.2.3 Reachability of Hybrid Dynamical Systems and Optimal Control Problem

Let  $X_0 \subset X$  be the set of allowed initial states, i.e. the set of states from which the system (5.13)-(5.14) is allowed to start. Define the reachable set as the set of all points attainable by  $y$  after a time  $s$  starting within the set of allowed initial states  $X_0$  to be

$$\begin{aligned} R_{X^0}(s) &= \{x \mid \exists q \in Q, y_{x,q;s} \in Y_{[0,s]}^{x,q}, y_{x,q;s}(0) \in X^0, \text{ and } y_{x,q;s}(\theta) \in K, \forall \theta \in [0, s]\} \\ &= \{x \mid \exists (q, p) \in Q \times P, y_{x,q,p;s} \in S_{[0,s]}^{x,q,p}, y_{x,q,p;s}(0) \in X^0 \\ &\quad \text{and } y_{x,q,p;s}(\theta) \in K, \forall \theta \in [0, s]\}. \end{aligned} \quad (5.18)$$

In other words, the reachable set  $R_{X^0}(s)$  contains the values of  $y_{x,q;s}(s)$ , regardless of the final discrete state, for all admissible trajectories – i.e., trajectories obtained through an admissible hybrid control – starting within the set of possible initial states  $X_0$ , that never leave set  $K$ .

Observe that (6.9) defines the reachable set  $R_{X^0}(s)$  in terms of both admissible trajectory sets  $Y$  and  $S$ .

**Remark 5.2.** *In particular, the information contained in (6.9) allows one to determine the first time where the reachable set is empty. More precisely, given  $X_0 \subset X$ , define  $s^* \geq 0$  to be*

$$s^* = \inf\{s \mid R_{X^0}(s) \subset \emptyset\}. \quad (5.19)$$

*The time (6.10) is the autonomy of the hybrid system (5.13)-(5.14). Indeed, one can readily see that if no more admissible energy states are attainable after  $s^*$ , any admissible trajectory must come to a stop beyond this time. Therefore,  $s^*$  is the longest time during which the state remains inside  $K$ .*

The following proposition ensures that the space of admissible trajectories is a compact set.

**Proposition 5.2.** *Given  $T' > 0$ , the admissible trajectory set  $Y_{[0,T']}^{x,q}$  is a compact set in  $C([0, T'])$  endowed with the topology  $W^{1,1}$ .*

*Proof.* Fix  $q \in Q$  and  $0 \leq s < t \leq T'$ . Consider a bounded admissible continuous control sequence  $u_n \in L^1([s, t])$ . Since  $u_n$  is bounded, there exists a subsequence  $u_{n_j}$  such that  $u_{n_j} \rightharpoonup u$  in  $L^1([s, t])$ . Invoking (H1) – (H2), we have  $y^{u_n} = y_n \rightharpoonup y$  in  $W^{1,1}([s, t])$ . Since  $W^{1,1}([s, t])$  is compactly embedded in  $C^0([0, t])$ , we get the strong convergence of the solution  $y_n \rightarrow y$  in  $C^0([s, t])$ . Hypothesis (H3) guarantees that the limit function  $y$  is a solution of (5.13). Because all controls  $u_n$  and the limit control  $u$  are admissible,  $y$  is an admissible solution.

So far, the proof shows that the limit trajectory is admissible when  $q$  is hold constant. Consider a sequence of admissible discrete control sequences  $(w)_n$  where the number of switching orders,  $0 \leq k_n \leq \lfloor T/\delta \rfloor$  may depend on  $n$ . Since each term of this sequence has a (first) discrete component and is bounded on the (second) continuous component, then, as  $n \rightarrow \infty$  there exists a subsequence  $(w)_{n_l}$  and  $\Lambda > 0$  such that  $q_{n_l} = q$  for all  $l > \Lambda$ . This implies  $k_n \rightarrow k$ . As the number of switches is constant from the  $\Lambda^{\text{th}}$  term and  $s, t$  are arbitrary, one can obtain, using the limit discrete control sequence  $w$ , the time intervals  $[s_{i-1}, s_i]$  over which  $q_i$  is constant and the argument in the first paragraph of the proof. Because the trajectory is continuous and admissible on all time intervals  $[s_{i-1}, s_i], i = 1, \dots, K$ , it is admissible on  $[0, T]$ .

Moreover, observe that  $\forall s \in [0, t], y_n(s) \in K$ . Since for all  $s \in [0, t], y_n(s) \rightarrow y(s)$ , by the compactness of  $K$  we get that  $\forall s \in [0, t], y(s) \in K$ , which completes the proof.  $\square$

**Remark 5.3.** *The arguments presented in the above proof can be slightly modified to show that the admissible trajectory set with fixed final  $p$ ,  $S_{[0,T']}^{x,q,p}$  is compact. Also in a similar way, the proof can be*



adapted to show that the reachable  $R_{X^0}$  is closed. Indeed, by the compactness of set  $X_0$ , a sequence of initial conditions  $(y_0)_n \in X_0$ , associated with admissible trajectories  $y_n \in Y_{[0,T']}^{x,q}$ , converges to  $y_0 \in X_0$  which is also the initial condition for the limiting trajectory  $y_n \rightarrow y$ .

In order to characterize the reachable set  $R_{X^0}$  this paper follows the classic level-set approach [64]. The idea is to describe (6.9) as the negative region of a function  $v$ . It is well known that the function  $v$  can be defined as the value function of some optimal control problem. In the case of system (5.13)-(5.14),  $v$  happens to be the value function of a hybrid optimal control problem.

Consider a Lipschitz continuous function  $\tilde{\phi} : X \rightarrow \mathbb{R}$  such that

$$\tilde{\phi}(x) \leq 0 \Leftrightarrow x \in X^0.$$

Such a function always exists – for instance, the signed distance function  $d_{X^0}$  from the set  $X^0$ . For  $L_K > 0$ , one can construct a bounded function  $\phi : X \rightarrow \mathbb{R}$  as

$$\phi(x) = \max(\min(\tilde{\phi}(x), L_K), -L_K). \quad (5.20)$$

For a given point  $s \geq 0$  and hybrid state vector  $(x, q, p) \in X \times Q \times P$ , define the value function to be

$$v_0(x, q, p, s) = \inf_{S_{[0,s]}^{x,q,p}} \{\phi(y_{x,q,p;s}(0)) \mid y_{x,q,p;s}(\theta) \in K, \forall \theta \in [0, s]\} \quad (5.21)$$

Observe that (6.9) works as a level-set to the negative part of (6.14). Indeed, since (6.14) contains only admissible trajectories that remain in  $K$ , by (6.13) implies that  $v_0(x, q, p, s)$  is negative if and only if  $y_{x,q,p;s}(0)$  is inside  $X_0$ , which in turn implies that  $x \in R_{X^0}(s)$ .

Remark however that when defining the value function with (6.14), one includes state constraints, with the condition that  $y_{x,q,p;s}(\theta) \in K$  for all times. When  $K \neq \mathbb{R}^d$ , one cannot expect  $v$  to be continuous and the HJ equation associated with (6.14) may have several solutions. In order to bypass such regularity issues, this paper follows the idea of [17, 16]. Define a Lipschitz continuous function  $\tilde{\varphi} : X \rightarrow \mathbb{R}$  to be

$$\tilde{\varphi}(x) \leq 0 \Leftrightarrow x \in K,$$

and

$$\varphi(x) = \max(\min(\tilde{\varphi}(x), L_K), -L_K). \quad (5.22)$$

Then, for a given  $s \geq 0$  and  $(x, q, p) \in X \times Q \times P$ , define a total penalization function to be

$$J(x, q, p, s; y) = \left( \phi(y_{x,q,p;s}(0)) \bigvee \max_{\theta \in [0,s]} \varphi(y_{x,q,p;s}(\theta)) \right)$$

and then, the optimal value :

$$v(x, q, p, s) = \inf_{y \in S_{[0,s]}^{x,q,p}} J(x, q, p, s; y). \quad (5.23)$$

Observe that (6.14) and (6.16) are bounded thanks to the constructions (6.13) and (6.15) respectively. The idea in place is that one needs only to look at the sign of  $v_0$  or  $v$  to obtain information about the reachable set. Therefore, the bound  $L_K$  removes the necessity of dealing with an unbounded value function besides providing a convenient value for numerical computations. In order to ensure that constructions (6.13) and (6.15) do not interfere with the original problem's formulation (in the sense that using  $\phi$ ,  $\varphi$  or  $\tilde{\phi}$ ,  $\tilde{\varphi}$  should yield the same results), given  $s$ ,  $X_0$  and  $x$ , assuming one does use signed distance functions to sets  $X_0, K$ , it suffices to take  $L_K > \sup_{x' \in X_0} x' e^{L_f |x-x'|s}$ , where  $L_f$  is the Lipschitz constant of  $f$ .

### 5.3 Main Results

The next proposition certifies that (6.9) is indeed a level-set of (6.14) and (6.16).

**Proposition 5.3.** *Assume (H1)-(H3). Define Lipschitz continuous functions  $\phi$  and  $\varphi$  by (6.13) and (6.15) respectively. Define value functions  $v_0$  and  $v$  by (6.14) and (6.16) respectively. Then, for  $s \geq 0$ , the reachable set is given by*

$$R_{X^0}(s) = \{x \mid \exists (q, p) \in Q \times P, v_0(x, q, p, s) \leq 0\} = \{x \mid \exists (q, p) \in Q \times P, v(x, q, p, s) \leq 0\} \quad (5.24)$$

*Proof.* The proof begins by showing that  $v_0(x, q, p, s) \leq 0 \Rightarrow v(x, q, p, s) \leq 0$ . Assume  $v_0(x, q, p, s) \leq 0$ . Then, using lemma 5.2, there exists an admissible trajectory such that

$$\phi(y_{x,q,p;s}(0)) \leq 0, \quad y_{x,q,p;s}(\theta) \in K, \quad \forall \theta \in [0, s].$$

Thus,  $\max_{\theta \in [0, s]} \varphi(y_{x,q,p;s}(\theta)) \leq 0$  and

$$v(x, q, p, s) \leq \max(\phi(y_{x,q,p;s}(0)), \max_{\theta \in [0, s]} \varphi(y_{x,q,p;s}(\theta))) \leq 0$$

Now, show that  $v(x, q, p, s) \leq 0 \Rightarrow v_0(x, q, p, s) \leq 0$ . Assume  $v(x, q, p, s) \leq 0$ . Then, by lemma 5.2 there exists a trajectory that verifies

$$\max(\phi(y_{x,q,p;s}(0)), \max_{\theta \in [0, s]} \varphi(y_{x,q,p;s}(\theta))) \leq 0.$$

By the definition of  $\varphi$ ,  $\forall \theta \in [0, s]$ ,

$$\max(\varphi(y_{x,q,p;s}(\theta))) \leq 0 \Rightarrow y_{x,q,p;s}(\theta) \in K,$$

which implies  $v_0(x, q, p, s) \leq 0$ . Therefore,  $u$  and  $v$  have the same negative regions.

Now, assume  $y_{x,q,p;s}(s) \in R_{X^0}(s)$ . Then, by definition, there exists  $(q, p) \in Q \times P$  and an admissible trajectory such that  $y_{x,q,p;s}(\theta) \in K$  for all time and  $y_{x,q,p;s}(0) \in X_0$ . This implies that  $\max_{\theta \in [0, s]} (\varphi(y_{x,q,p;s}(\theta))) \leq 0$  and  $\phi(y_{x,q,p;s}(0)) \leq 0$ . It follows that  $v(x, q, p, s) \leq J(x, q, p, s; y) \leq 0$ .

Conversely, assume  $v(x, q, p, s) \leq 0$ . For any optimal trajectory  $\hat{y}$  (which is admissible thanks to proposition 5.2)  $v(x, q, p, s) = J(x, q, p, s; \hat{y}) \leq 0$ . Since the maximum of the two quantities is non positive only if they are both non positive one can draw the desired conclusion.  $\square$

Proposition (6.1) sets the equivalence between (6.9) and the negative regions of (6.14) and (6.16). In particular, it states that it suffices to compute  $v$  or  $v_0$  in order to obtain information about  $R_{X^0}$ . In this sense, this paper focuses on (6.16), which is associated with an optimal control problem with no state constraints.

In the sequel, it is shown that (6.16) is the unique (viscosity) solution of a quasi-variational inequalities' system. The first step is to state a dynamic programming principle for (6.16).

First, we present some preliminary notation. Given  $t > 0$ , set  $T = (0, t]$ ,  $\Omega = X \times Q \times P \times T$  and denote its closure by  $\bar{\Omega}$ . For a fixed  $p_0 \in P$ , define  $\Omega|_{p_0} = X \times Q \times \{p_0\} \times T$  and denote  $\bar{\Omega}|_{p_0}$  the closure of  $\Omega|_{p_0}$ . Define

$$\mathcal{V}(\bar{\Omega}) := \{v \mid v : \bar{\Omega} \rightarrow \mathbb{R}, v \text{ bounded}\}, \quad (5.25)$$

$$\mathcal{V}(\bar{\Omega}|_{p_0}) := \{v \mid \text{for } p_0 \in P, v : \bar{\Omega}|_{p_0} \rightarrow \mathbb{R}, v \text{ bounded}\} \quad (5.26)$$

For  $v \in \mathcal{V}(\bar{\Omega})$ , denote its upper and lower envelope at point  $(x, q, p, s) \in \bar{\Omega}$  respectively as  $v^*$  and  $v_*$ :

$$v^*(x, q, p, s) = \limsup_{\substack{x_n \rightarrow x \\ q_n \rightarrow q \\ p_n \rightarrow p \\ s_n \rightarrow s}} v(x_n, q_n, p_n, s_n) \quad (5.27)$$

$$v_*(x, q, p, s) = \liminf_{\substack{x_n \rightarrow x \\ q_n \rightarrow q \\ p_n \rightarrow p \\ s_n \rightarrow s}} v(x_n, q_n, p_n, s_n) \quad (5.28)$$

In the case where  $p_0 \in P$  is fixed and  $v \in \mathcal{V}(\bar{\Omega}|_{p_0})$ , the upper and lower envelopes of  $v$  are also given by (5.27), (5.28) with  $p_n = p_0$  for all  $n$ .

Now, fix  $p = 0$  and define the non-local switch operators  $M, M^+, M^- : \mathcal{V}(\bar{\Omega}|_0) \rightarrow \mathcal{V}(\bar{\Omega}|_0)$  to be

$$\begin{aligned} (Mv)(x, q, 0, s) &= \inf_{\substack{w \in W(q) \\ p' \geq \delta}} v(x, g(w, q), p', s) \\ (M^+v)(x, q, 0, s) &= \inf_{\substack{w \in W(q) \\ p' \geq \delta}} v^*(x, g(w, q), p', s) \\ (M^-v)(x, q, 0, s) &= \inf_{\substack{w \in W(q) \\ p' \geq \delta}} v_*(x, g(w, q), p', s) \end{aligned}$$

The action of these operators on the value function represents a switch that respects the lag constraint. They operate whenever a switch is activated, which is equivalent to the condition  $p = 0$ . Therefore, they are defined only for a fixed  $p = 0$ . Let us recall here some classical properties of operators  $M, M^+$  and  $M^-$  (adapted from [90]):

**Lemma 5.2.** *Let  $v \in B\mathcal{V}(\bar{\Omega})$ . Then  $M^+v^* \in BUSC(\bar{\Omega})$  and  $M^-v_* \in BLSC(\bar{\Omega})$ . Moreover  $(Mv)^* \leq M^+v^*$  and  $(Mv)_* \geq M^-v_*$ .*

*Proof.* Fix  $q \in Q$ ,  $p = 0$  and  $\epsilon > 0$ . Let  $w^* \in W(x, q)$  and  $p^* > 0$  be such that for all  $x \in X$  and  $s \in T$ ,  $(M^+v^*)(x, q, p, s) \geq v^*(x, g(w^*, q), p^*, s) - \epsilon$ . Consider sequences  $x_n \rightarrow x$  and  $s_n \rightarrow s$ . Then

$$\begin{aligned} M^+v^*(x, q, 0, s) &\geq v^*(x, g(w^*, q), p^*, s) - \epsilon \\ &\geq \limsup_{\substack{x_n \rightarrow x \\ s_n \rightarrow s}} v^*(x_n, g(w^*, q), p^*, s_n) - \epsilon \\ &\geq \limsup_{\substack{x_n \rightarrow x \\ s_n \rightarrow s}} \inf_{w \in W(x_n, q), p' \geq \delta} v^*(x_n, g(w, q), p', s_n) - \epsilon \\ &= \limsup_{\substack{x_n \rightarrow x \\ s_n \rightarrow s}} (M^+v^*)(x_n, q, 0, s_n) - \epsilon. \end{aligned}$$

Notice that  $q$  and  $p$  are held constant throughout the inequalities and thus, the limsup of the jump operator considering only sequences  $x_n \rightarrow x$  and  $s_n \rightarrow s$  corresponds to its envelope at the limit point. Then, by the arbitrariness of  $\epsilon$ , this proves the upper semi-continuity of  $M^+v^*$ . The lower semi-continuity of  $M^-v_*$  can be obtained in a similar fashion.

Now, observe that  $Mv \leq M^+v^*$ . Taking the upper envelope of each side, one obtains:

$$(Mv)^* \leq (M^+v^*)^* = M^+v^*.$$

By the same kind of reasoning  $Mv \geq M^-v_*$  and

$$(Mv)_* \geq (M^-v_*)_* = M^-v_*.$$

□

The next proposition is the dynamic programming principle verified by (6.16):

**Proposition 5.4.** *The value function (6.16) satisfies the following dynamic programming principle:*

(i) For  $s=0$ ,

$$v(x, q, p, 0) = \max(\phi(x), \varphi(x)), \quad \forall (x, q, p) \in X \times Q \times P, \quad (5.29)$$

(ii) For  $p=0$ ,

$$v(x, q, 0, s) = (Mv)(x, q, 0, s), \quad (x, q, s) \in X \times Q \times T, \quad (5.30)$$

(iii) For  $(x, q, p, s) \in \Omega$ , define the non-intervention zone as  $\Sigma = (0, p \wedge s)$ . Then, for  $h \in \Sigma$ ,

$$v(x, q, p, s) = \inf_{S_{[s-h, s]}^{x, q, p}} \left\{ v(y_{x, q, p; h}(s-h), q, p-h, s-h) \bigvee \max_{\theta \in [s-h, s]} \varphi(y_{x, q, p; h}(\theta)) \right\} \quad (5.31)$$

*Proof.* The dynamic programming principle is composed of three parts.

(i): Equality (6.20) is obtained directly by definition (6.16).

(ii): " $\leq$ ". Let  $(x, q, s) \in X \times Q \times T$  and  $p=0$ . Consider a hybrid control  $a = (u(\cdot), \{w_i, s_i\}_{i=1}^N)$  and an associated trajectory  $y^a$ . Construct a control  $\bar{a} = (\bar{u}(\cdot), \{\bar{w}_i, \bar{s}_i\}_{i=1}^{N-1})$  with associated trajectory  $y^{\bar{a}}$ , where  $\bar{u} = u$ ,  $\bar{w}_i = w_i$  and  $\bar{s}_i = s_i$  for  $i = 1, \dots, N-1$  and  $s_N = s$ ,  $w_N = w'$ . Then, one obtains,

$$\begin{aligned} v(x, q, 0, s) &\leq J(x, q, 0, s; y^a) \\ &= J(x, g(w', q), p', s; y^{\bar{a}}), \end{aligned}$$

where the controller must respect the condition  $p' \geq \delta$  for it to be admissible. Since  $\bar{a}$  is arbitrary, one can choose it such that

$$\begin{aligned} v(x, q, 0, s) &\leq \inf_{y^{\bar{a}} \in S_{[0, s]}^{(x, g(w', q), p')}} J(x, g(w', q), p', s; y^{\bar{a}}) + \epsilon_1 \\ &= v(x, g(w', q), p', s) + \epsilon_1, \end{aligned}$$

where  $\epsilon_1 > 0$ . Now, choose the last switch  $w'$  and  $p'$  such that

$$\begin{aligned} v(x, q, 0, s) &\leq \inf_{\substack{w' \in W(q) \\ p' \geq \delta}} v(x, g(w', q), p', s) + \epsilon_1 \\ &= (Mv)(x, q, p, s) + \epsilon_1. \end{aligned}$$

" $\geq$ ". For  $(x, q, p) \in X \times Q \times P$  and  $p=0$ , there always exists an admissible control that  $a_\epsilon$ , such that there exists  $\epsilon_2 > 0$  and,

$$v(x, q, 0, s) + \epsilon_2 \geq J(x, q, 0, s; y^{a_\epsilon})$$

Using the same hybrid control constructions as in the " $\leq$ " case, one obtains

$$\begin{aligned} J(x, q, 0, s; y^{a_\epsilon}) &= J(x, g(w', q), p', s; y^{a_\epsilon}) \\ &\geq v(x, g(w', q), p', s) \\ &\geq \inf_{\substack{w' \in W(q) \\ p' \geq \delta}} v(x, g(w', q), p', s) \\ &= (Mv)(x, q, 0, s). \end{aligned}$$

Relation (6.21) is obtained by the arbitrariness of both  $\epsilon_1, \epsilon_2$ .

(iii): " $\leq$ ". For  $(x, q, p, s) \in \Omega$  and  $0 < h \leq p \wedge s$ , (6.16) yields

$$v(x, q, p, s) \leq \max \left( \left( \phi(y_{x,q,p;s}(0)) \bigvee_{\theta \in [0, s-h]} \varphi(y_{x,q,p;s}(\theta)) \right), \max_{\theta \in [s-h, s]} \varphi(y_{x,q,p;s}(\theta)) \right), \quad (5.32)$$

for any  $y \in S_{[0,s]}^{x,q,p}$ . By the choice of  $h$ , there is no switching between times  $s-h$  and  $s$ . Write the admissible control  $a = (u, w)$  as  $a_0 = (u_0, w)$  and  $a_1 = (u_1, w)$  with

$$\begin{aligned} u_0(s) &= u(s), \quad s \in [0, s-h], \\ u_1(s) &= u(s), \quad s \in (s-h, s]. \end{aligned}$$

Since  $a$  is admissible, both controls  $a_0, a_1$  are also admissible. Denote the trajectory associated with controls  $a, a_0, a_1$  respectively by  $y^a, y^0, y^1$ . Then,  $y^a \in S_{[0,s]}^{x,q,p}$  and by continuity of the trajectory we achieve the following decomposition:

$$y^1 \in S_{[s-h,s]}^{x,q,p}, \quad y^0 \in S_{[0,s-h]}^{y^1(s-h),q,p-h}.$$

The above decomposition together with inequality (5.32) yields

$$v(x, q, p, s) \leq \max \left( \left( \phi(y^0(0)) \bigvee_{\theta \in [0, s-h]} \varphi(y^0(\theta)) \right), \max_{\theta \in [s-h, s]} \varphi(y^1(\theta)) \right),$$

And one concludes after minimizing with respect to the trajectories associated with  $a_0$  and  $a_1$ .

The " $\geq$ " part uses a particular  $\epsilon$ -optimal controller and the same decomposition, allowing to conclude by the arbitrariness of  $\epsilon$ . This is possible because there is no switching between  $s-h$  and  $s$ .

□

A direct consequence of proposition 5.4 is the Lipschitz continuity of the value function, stated in the next proposition:

**Proposition 5.5.** *Assume (H1)-(H2). Define Lipschitz continuous functions  $\phi$  and  $\varphi$  by (6.13) and (6.15), with Lipschitz constants  $L_\phi$  and  $L_\varphi$  respectively. Then, for  $p > 0$ , (6.16) is Lipschitz continuous.*

*Proof.* Fix  $s, s' > 0$ ,  $x, x' \in X$ ,  $q \in Q$ ,  $p > 0$ . Then, using  $\max(A, B) - \max(C, D) \leq \max(A - B, C - D)$ , one obtains,

$$\begin{aligned} |v(x, q, p, s) - v(x', q, p, s)| &\leq \max \left( |\phi(y_{x,q,p;s}(0)) - \phi(y_{x',q,p;s}(0))|, \right. \\ &\quad \left. \max_{\theta \in [0, s]} |\varphi(y_{x,q,p;s}(\theta)) - \varphi(y_{x',q,p;s}(\theta))| \right) \\ &\leq \max \left( L_\phi |y_{x,q,p;s}(0) - y_{x',q,p;s}(0)|, \right. \\ &\quad \left. L_\varphi \max_{\theta \in [0, s]} (|y_{x,q,p;s}(\theta) - y_{x',q,p;s}(\theta)|) \right) \\ &\leq L_v |x - x'|, \end{aligned}$$

where  $L_v = \max(L_\phi, L_\varphi)e^{L_f s}$ .

Now, take  $h > 0$  and observe that  $v(x, q, p, s) \geq \varphi(x)$ . Then,

$$\begin{aligned}
|v(x, q, p+h, s+h) - v(x, q, p, s)| &\leq \max \left( \left| v(y_{x,q,p+h;s+h}(s), q, p, s) - v(x, q, p, s) \right|, \right. \\
&\quad \left. \left| \max_{\theta \in [s, s+h]} \varphi(y_{x,q,p;s+h}(\theta)) - \varphi(x) \right| \right) \\
&\leq \max \left( L_v \left| y_{x,q,p+h;s+h}(s) - y_{x,q,p;s}(s) \right|, \right. \\
&\quad \left. L_\varphi \max_{\theta \in [s, s+h]} |y_{s,q,p+h;s+h}(\theta) - y_{x,q,p;s}(s)| \right) \\
&\leq L_f \max(L_v, L_\varphi) h.
\end{aligned}$$

□

In order to proceed to the HJB equations, define the Hamiltonian to be

$$H(s, x, q, z) = \sup_{u \in U(q)} f(s, x, u, q) \cdot z \quad (5.33)$$

Before stating the next result, we recall the notion of viscosity solution [30] used throughout this paper.

**Definition 5.2.** A function  $u_1$  (resp.  $u_2$ ) upper semi-continuous (u.s.c.) (resp. lower semi-continuous (l.s.c)) is a viscosity subsolution (resp. supersolution) if there exists a continuously differentiable function  $\psi$  such that  $u_1 - \psi$  has a local maximum (resp.  $u_2 - \psi$  has a local minimum) at  $(x, q, p, s) \in \bar{\Omega}$  and

$$\partial_s \psi + \partial_p \psi + H(s, x, q, \nabla_x \psi) \bigwedge u_1 - \varphi(x) \leq 0 \quad \text{if } (x, q, p, s) \in \Omega \quad (5.34)$$

$$u_1(x, q, p, s) \leq (M^+ u_1)(x, q, p, s) \quad \text{if } p = 0, \quad (5.35)$$

$$u_1(x, q, p, s) \leq \max(\phi(x), \varphi(x)) \quad \text{if } s = 0, \quad (5.36)$$

(with the inequalities signs inversed and  $M^-$  instead of  $M^+$  for  $u_2$ ). A bounded function  $u$  is a (viscosity) solution of (5.34)–(5.36) if  $u^*$  is a subsolution and  $u_*$  is a supersolution.

The next statement shows that the value function defined in (6.16) is a solution of a quasi-variational system.

**Theorem 5.1.** Assume (H1)–(H5). Let the Lipschitz functions  $\phi$  and  $\varphi$  be defined by (6.13) and (6.15) respectively. Then, the Lipschitz, bounded value function  $v$  defined in (6.16) is a viscosity solution of the quasi-variational inequality

$$\partial_s v + \partial_p v + H(s, x, q, \nabla_x v) \bigwedge v - \varphi(x) = 0 \quad , \quad \forall (x, q, p, s) \in \Omega, \quad (5.37)$$

$$v(x, q, 0, s) = (Mv)(x, q, 0, s) \quad , \quad \forall (x, q, s) \in X \times Q \times [0, \infty[ \quad (5.38)$$

$$v(x, q, p, 0) = \max(\phi(x), \varphi(x)) \quad , \quad \forall (x, q, p) \in X \times Q \times P. \quad (5.39)$$

*Proof.* By definition,  $v$  satisfies the initial condition (5.39). The boundary condition (5.38) is deduced from proposition 5.4. Now, we proceed to show that (i)  $v$  is a supersolution and (ii) a subsolution of (5.37):

First, let us prove the supersolution property (i). To satisfy  $\min(A, B) \geq 0$  one needs to show  $A \geq 0$  and  $B \geq 0$ . Since  $v - \varphi(x) \geq 0$ , it is immediate that  $B \geq 0$ . Now, consider  $0 < h \leq p \wedge s$ . Let  $\psi$  be a continuously differentiable function such that  $v - \psi$  attains a minimum at  $(x, q, p, s)$ . Then, using proposition (5.4)(iii) and selecting an  $\epsilon$ -optimal controller, dependent on  $h$ , with associated trajectory  $y_{x,q,p;s}^\epsilon$ , it follows that

$$\begin{aligned} \psi(x, q, p, s) = v(x, q, p, s) &\geq \inf_{S_{[0,s]}^{x,q,p}} v(y_{x,q,p;s}(s-h), q, p-h, s-h) \\ &\geq v(y_{x,q,p;s}^\epsilon(s-h), q, p-h, s-h) - h\epsilon \\ &= \psi(y_{x,q,p;s}^\epsilon(s-h), q, p-h, s-h) - h\epsilon \end{aligned}$$

and then,

$$\psi(x, q, p, s) - \psi(y_{x,q,p;s}^\epsilon(s-h), q, p-h, s-h) \geq -h\epsilon.$$

Since the control domain is bounded and using the continuity of  $f, p$  and  $\psi$  we divide by  $h$  and take the limit  $h \rightarrow 0$  to obtain

$$\partial_s \psi + \partial_p \psi + H(s, x, q, \nabla_x \psi) \geq -\epsilon$$

and conclude that  $A \geq 0$  by the arbitrariness of  $\epsilon$ .

For (ii), observe that for  $\min(A, B) \leq 0$  it suffices to show that  $A \leq 0$  or  $B \leq 0$ . If  $v(x, q, p, s) = \varphi(x)$ , it implies  $B \leq 0$ . On the contrary, if  $v(x, q, p, s) > \varphi(x)$ , then there exists a  $\Sigma \ni h \geq 0$  small enough so that

$$v(y_{x,q,p;s}^u(s-h), q, p-h, s-h) > \max_{\theta \in [s-h, h]} \varphi(y_{x,q,p;s}^u(\theta))$$

strictly, using the Lipschitz continuity of  $f, p$  and the compactness of  $U$  (which ensures the trajectories will remain near each other). Thus, proposition 5.4(iii) yields

$$v(x, q, p, s) = \inf_{y^u \in S_{[s-h, s]}^{x,q,p}} v(y_{x,q,p;s}^u(s-h), q, p-h, s-h).$$

Fix an arbitrary  $u \in U$  and consider a constant control  $u(s) = u$  for  $0 < s < h$ . Let  $\psi$  be a continuously differentiable function such that  $v - \psi$  attains a maximum at  $(x, q, p, s)$ . Also, without loss of generality, assume that  $v(x, q, p, s) = \psi(x, q, p, s)$ . Hence,

$$\begin{aligned} v(x, q, p, s) &\leq v(y_{x,q,p;s}^u(s-h), q, p-h, s-h) \\ &\leq \psi(y_{x,q,p;s}^u(s-h), q, p-h, s-h) \end{aligned}$$

and by dividing by  $h$  and taking  $h \rightarrow 0$  one obtains

$$\partial_s \psi + \partial_p \psi + f(s, x, u) \cdot \nabla_x \psi \leq 0.$$

Since  $u$  is arbitrary and admissible, we conclude that  $A \leq 0$ , which completes the proof.  $\square$

Theorem 5.1 provides a convenient way to characterize the value function whose level-set is the reachable set defined in (6.9). However, in order to be sure that the solution that stems from (5.37)-(5.39) corresponds to (6.16), a uniqueness result is necessary. This is achieved by a comparison principle which is stated in the next theorem. Let  $BUSC(\Omega)$  and  $BLSC(\Omega)$  respectively be the space of u.s.c. and l.s.c. functions defined over the set  $\Omega$ .

**Theorem 5.2.** *Let  $u_1 \in BUSC(\Omega)$  and  $u_2 \in BLSC(\Omega)$  be, respectively, sub- and supersolution of*

$$\partial_s u + \partial_p u + H(s, x, q, \nabla_x u) \bigwedge u - \varphi(x) = 0 \quad , \quad \forall (x, q, p, s) \in \Omega, \quad (5.40)$$

$$u(x, q, 0, s) - (Mu)(x, q, 0, s) = 0 \quad , \quad \forall (x, q, s) \in X \times Q \times [0, \infty[ \quad (5.41)$$

$$u(x, q, p, 0) = \max(\phi(x), \varphi(x)) \quad , \quad \forall (x, q, p) \in X \times Q \times P. \quad (5.42)$$

Then,  $u_1 \leq u_2$  in  $\bar{\Omega}$ .

The proof is inspired by earlier work on uniqueness results for hybrid control problems. The idea is to show that  $u_1 \leq u_2$  in all domain  $\Omega$  and then on the boundary  $p = 0$ . The main difficulty arises when dealing with points in the boundary  $p = 0$  where the system has a switching condition given by a non-local switch operator. This is tackled by the utilization of “friendly giant”-like test functions [9], [55]. Classically, these functions are used to prove uniqueness for elliptic problems with unbounded value functions where they serve to localize some arguments regardless of the function’s possible growth at infinity. This feature proves itself very useful in our case because one can properly split the domain in no-switching and switching regions. In this work, the lag condition for the switch serves as an equivalent to the “landing condition”— which states that after an autonomous switch the system must land at some positive distance away from the autonomous switch set [32], [90].

*Proof.* Let  $\Omega$  be defined as above,  $\partial\Omega|_T = X \times Q \times P \times \{0\}$  and  $\partial\Omega|_P = X \times Q \times \{0\} \times T$ .

First, the comparison principle is proved for  $\partial\Omega|_T$  (case 1), followed by  $\Omega$  (case 2) and finally for  $\partial\Omega|_P$  (case 3), which concludes the proof for  $\bar{\Omega}$ .

Case 1: At a point  $(x, q, p, t) \in \partial\Omega|_T$ , from the sub- and supersolution properties,

$$\begin{aligned} u_1(x, q, p, 0) - \max(\phi(x), \varphi(x)) &\leq 0, \\ -u_2(x, q, p, 0) + \max(\phi(x), \varphi(x)) &\leq 0, \end{aligned}$$

which readily yields  $u_1 \leq u_2$  in  $\partial\Omega|_T$ .

Case 2: Start by using to sub- and supersolution properties of  $u_1, u_2$  to obtain, in  $\Omega$ ,

$$\min(\partial_s u_1 + \partial_p u_1 + H(s, x, q, \nabla_x u_1), u_1 - \varphi(x)) \leq 0, \quad (5.43)$$

$$\min(\partial_s u_2 + \partial_p u_2 + H(s, x, q, \nabla_x u_2), u_2 - \varphi(x)) \geq 0. \quad (5.44)$$

Expression (5.44) implies that both

$$u_2 \geq \varphi(x) \quad (5.45)$$

and

$$\partial_s u_2 + \partial_p u_2 + H(s, x, q, \nabla_x u_2) \geq 0. \quad (5.46)$$

From (5.43), one has to consider two possibilities. The first one is when  $u_1 \leq \varphi(x)$ . If so, together with (5.45), one has immediately  $u_1 \leq u_2$ . Now, if  $\partial_s u_1 + \partial_p u_1 + H(s, x, q, \nabla_x u_1) \leq 0$ , one turns to (5.46).

Define  $v = u_1 - u_2$ . Notice that  $v \in BUSC(\Omega)$ . The next step is to show that  $v$  is a subsolution of

$$\partial_s v + \partial_p v + H(s, x, q, \nabla_x v) = 0 \quad (5.47)$$

at  $(\bar{x}, \bar{q}, \bar{p}, \bar{s})$ .

Let  $\psi \in C^2(\Omega)$ , bounded, be such that  $v - \psi$  has a strict local maximum at  $(\bar{x}, \bar{q}, \bar{p}, \bar{s}) \in \Omega$ . Define auxiliary functions over  $\Omega^i \times \Omega^i$ ,  $i = 0, 1$  as

$$\begin{aligned} \Phi_\epsilon^i(x, p, s, \xi, \pi, \varsigma) &= u_1(x, i, p, s) - u_2(\xi, i, \pi, \varsigma) - \psi(x, p, s) \\ &\quad - \frac{|x - \xi|^2}{2\epsilon} - \frac{|p - \pi|^2}{2\epsilon} - \frac{|s - \varsigma|^2}{2\epsilon}. \end{aligned} \quad (5.48)$$

Because the boundedness of  $\psi$ ,  $u_1$  and  $u_2$  the suprema points are finite, for each  $i = 0, 1$ . Denote  $(x_\epsilon, p_\epsilon, s_\epsilon, \xi_\epsilon, \pi_\epsilon, \varsigma_\epsilon) \in \Omega^{\bar{q}} \times \Omega^{\bar{q}}$  a point such that

$$\Phi_\epsilon^{\bar{q}}(x_\epsilon, p_\epsilon, s_\epsilon, \xi_\epsilon, \pi_\epsilon, \varsigma_\epsilon) = \sup_{\Omega^{\bar{q}} \times \Omega^{\bar{q}}} \Phi_\epsilon^{\bar{q}}(x, p, s, \xi, \pi, \varsigma).$$

The following lemma establishes some estimations needed further in the proof:



**Lemma 5.3.** Define  $\Phi_\epsilon^i$  and  $(x_\epsilon, p_\epsilon, s_\epsilon, \xi_\epsilon, \pi_\epsilon, \varsigma_\epsilon)$  as above. Then, as  $\epsilon \rightarrow 0$ ,

$$\begin{aligned} \frac{|x_\epsilon - \xi_\epsilon|^2}{\epsilon} &\rightarrow 0, \quad \frac{|p_\epsilon - \pi_\epsilon|^2}{\epsilon} \rightarrow 0, \quad \frac{|s_\epsilon - \varsigma_\epsilon|^2}{\epsilon} \rightarrow 0, \\ |x_\epsilon - \xi_\epsilon| &\rightarrow 0, \quad |p_\epsilon - \pi_\epsilon| \rightarrow 0, \quad |s_\epsilon - \varsigma_\epsilon| \rightarrow 0, \end{aligned}$$

and  $(x_\epsilon, p_\epsilon, s_\epsilon, \xi_\epsilon, \pi_\epsilon, \varsigma_\epsilon) \rightarrow (\bar{x}, \bar{p}, \bar{s}, \bar{x}, \bar{p}, \bar{s})$

*Proof.* Writing

$$2\Phi_\epsilon^i(x_\epsilon, p_\epsilon, s_\epsilon, \xi_\epsilon, \pi_\epsilon, \varsigma_\epsilon) \geq \Phi_\epsilon^i(x_\epsilon, p_\epsilon, s_\epsilon, x_\epsilon, p_\epsilon, s_\epsilon) + \Phi_\epsilon^i(\xi_\epsilon, \pi_\epsilon, \varsigma_\epsilon, \xi_\epsilon, \pi_\epsilon, \varsigma_\epsilon),$$

for  $i = 0, 1$ , one obtains,

$$\begin{aligned} \frac{|x - \xi|^2}{\epsilon} + \frac{|p - \pi|^2}{\epsilon} + \frac{|s - \varsigma|^2}{\epsilon} &\leq (u_1 + u_2)(x_\epsilon, i, p_\epsilon, s_\epsilon) - (u_1 + u_2)(\xi_\epsilon, i, \pi_\epsilon, \varsigma_\epsilon) + \\ &\quad \psi(x_\epsilon, p_\epsilon, s_\epsilon) - \psi(\xi_\epsilon, \pi_\epsilon, \varsigma_\epsilon), \end{aligned}$$

which means that, since  $\psi$ ,  $u_1$  and  $u_2$  are bounded that

$$\frac{|x_\epsilon - \xi_\epsilon|^2}{\epsilon} \leq C^*, \quad \frac{|p_\epsilon - \pi_\epsilon|^2}{\epsilon} \leq C^*, \quad \frac{|s_\epsilon - \varsigma_\epsilon|^2}{\epsilon} \leq C^*, \quad (5.49)$$

where  $C^*$  depends on the sup  $|u_1|$ , sup  $|u_2|$ , sup  $|\psi|$  and is independent of  $\epsilon$ . Expression (5.49) yields

$$|x_\epsilon - \xi_\epsilon| \leq \sqrt{\epsilon C^*}, \quad |p_\epsilon - \pi_\epsilon| \leq \sqrt{\epsilon C^*}, \quad |s_\epsilon - \varsigma_\epsilon| \leq \sqrt{\epsilon C^*}.$$

which implies that the doubled terms tend to zero.

Since  $(\bar{x}, \bar{q}, \bar{p}, \bar{s})$  is a strict maximum of  $v - \psi$ , one gets  $(x_\epsilon, p_\epsilon, s_\epsilon, \xi_\epsilon, \pi_\epsilon, \varsigma_\epsilon) \rightarrow (\bar{x}, \bar{p}, \bar{s}, \bar{x}, \bar{p}, \bar{s})$ . Remark that, since  $\bar{p} > 0$ , one can always choose a suitable subsequence  $\epsilon_n \rightarrow 0$  such that all  $p_{\epsilon_n} > 0$ , avoiding thus touching the switching boundary.  $\square$

A straightforward calculation allows to show that there exists  $a, b \in \mathbb{R}$  such that

$$\begin{aligned} (a, b, D_\epsilon) &\in D^- u_2(\xi_\epsilon, \bar{q}, \pi_\epsilon, \varsigma_\epsilon) \\ (a + \partial_s \psi, b + \partial_p \psi, D_\epsilon + \nabla_x \psi) &\in D^+ u_1(x_\epsilon, \bar{q}, p_\epsilon, s_\epsilon), \end{aligned}$$

where  $D^-, D^+$  respectively denote the sub- and super differential [30] and  $D_\epsilon = 2|x_\epsilon - \xi_\epsilon|/\epsilon$ , which implies

$$\begin{aligned} a + b + H(\varsigma_\epsilon, \xi_\epsilon, \bar{q}, D_\epsilon) &\geq 0 \\ a + \partial_s \psi + b + \partial_p \psi + H(s_\epsilon, x_\epsilon, \bar{q}, D_\epsilon + \nabla_x \psi) &\leq 0, \end{aligned}$$

which in turn yields, as  $\epsilon \rightarrow 0$ ,

$$\partial_s \psi + \partial_p \psi - L_f |\nabla_x \psi| \leq 0$$

at  $(\bar{x}, \bar{q}, \bar{p}, \bar{s}) \in \Omega$ . By adequately choosing the test functions  $\psi$ , one can repeat the arguments to show that this assertion holds for any point in  $\Omega$ . Thus, this establishes that  $v$  is a subsolution of (5.47) in  $\Omega$ .

Now, take  $\kappa > 0$  and define a non-decreasing differentiable function  $\chi_\kappa : (-\infty, 0) \rightarrow \mathbb{R}^+$  such that

$$\chi_\kappa(x) = 0, \quad x \leq -\kappa; \quad \chi_\kappa(x) \rightarrow \infty, \quad x \rightarrow 0.$$

Take  $\eta > 0$ , and define a test function

$$\nu(x, p, s) = \eta s^2 + \chi_\kappa(-p).$$

Observe that  $v - \nu$  achieves a maximum at a finite point  $(x_0, \bar{q}, p_0, s_0) \in \Omega$ . Since  $\kappa$  can be made arbitrarily small one can consider  $p_0 > \kappa$  without loss of generality. Therefore, using the subsolution property of  $v$ , by a straightforward calculation one has

$$2\eta s_0 \leq 0,$$

since  $\chi'_\eta(-p_0) = 0$ . The above inequality implies that  $s_0 = 0$ . Noticing that  $\nu(x_0, p_0, s_0) = v(x_0, \bar{q}, p_0, s_0) = 0$ , it follows

$$v(x, \bar{q}, p, s) \leq \eta s^2 + \chi_\kappa(-p)$$

for all  $s \in T$ ,  $x \in X$  and  $p > \kappa$ . Letting  $\eta \rightarrow 0$ ,  $\kappa \rightarrow 0$  and from the arbitrariness of  $\bar{q}$ , we conclude that  $v \leq 0$  in  $\Omega$ .

Case 3: In this case the switch lock variable arrives at the boundary of the domain, incurring thus a switch, as all others variables remain inside the domain. For all  $(x_0, q_0, p_0, s_0) \in \partial\Omega|_P$ , for any  $p \geq \delta$  one has (using case 2 and noticing that  $M^+u_1 = Mu_1$  and  $M^-u_2 = Mu_2$ )

$$(M^+u_1)(x_0, q_0, p_0, s_0) \leq u_1(x_0, q_0, p, s_0) \leq u_2(x_0, q_0, p, s_0).$$

Taking the infimum with respect to  $p$ , the above expression yields  $M^+u_1 \leq M^-u_2$  in  $\partial\Omega|_P$ . This suffices to conclude, since that by the sub- and supersolution properties

$$v = u_1 - u_2 \leq M^+u_1 - M^-u_2.$$

□

## 5.4 Numerical Analysis

### 5.4.1 Numerical Scheme and Convergence

Equations (5.37)-(5.39) can be solved using a finite differences scheme. This section proposes a class of discretization schemes and shows its convergence using the Barles-Souganidis [10] framework.

Set mesh sizes  $\Delta x > 0$ ,  $\Delta p > 0$ ,  $\Delta t > 0$  and denote the discrete grid point by  $(x_I, p_k, s_n)$ , where  $x_I = I\Delta x$ ,  $p_k = k\Delta p$  and  $s_n = n\Delta t$ , with  $I \in \mathbb{Z}^d$  and  $k, n$  integers. The approximation of the value function is denoted

$$v(x_I, q, p_k, s_n) = {}^qv_{Ik}^n$$

and the penalization functions are denoted  $\phi(x_I) = \phi_I$ ,  $\varphi(x_I) = \varphi_I$ . Define the following grids:

$$\begin{aligned} G^\# &= I\Delta x \times Q \times \Delta p\{0, 1, \dots, n_p\} \times \Delta t\{0, 1, \dots, n_s\}, \\ G_H^\# &= \Delta t\{0, 1, \dots, n_s\} \times I\Delta x \times Q \end{aligned}$$

and the discrete space gradient at point  $x_I$  for any general function  $\mu$ :

$$D^\pm \mu(x_I) = D^\pm \mu_I = \left( D_{x_1}^\pm \mu_I, \dots, D_{x_d}^\pm \mu_I \right),$$

where

$$D_{x_j}^\pm \mu_I = \pm \frac{\mu_{I^{j,\pm}} - \mu_I}{\Delta x},$$

with

$$I^{j,\pm} = (i_1, \dots, i_{j-1}, i_j \pm 1, \dots, i_d).$$

Define a numerical Hamiltonian  $\mathcal{H} : G_H^\# \times \mathbb{R}^d \times \mathbb{R}^d \rightarrow \mathbb{R}$  destined to be an approximation of  $H$ . We assume that  $\mathcal{H}$  verifies the following hypothesis:

(H6) There exists  $L_{H_1}, L_{H_2} > 0$  such that, for all  $s, x, q \in G_H^\#$  and  $A^+, A^-, B^+, B^- \in \mathbb{R}^d$ ,

$$\begin{aligned} |\mathcal{H}(s, x, q, A^+, A^-) - \mathcal{H}(s, x, q, B^+, B^-)| &\leq L_{H_1}(\|A^+ - B^+\| + \|A^- - B^-\|) \\ \|\mathcal{H}(s, x, q, A^+, A^-)\| &\leq L_{H_2}(\|A^+ + A^-\|). \end{aligned}$$

(H7) The Hamiltonian satisfies the monotonicity condition for all  $s, x, q \in G_H^\#$  and almost every  $A^+, A^- \in \mathbb{R}^d$ :

$$\partial_{A_i^+} \mathcal{H}(s, x, q, A^+, A^-) \leq 0, \text{ and } \partial_{A_i^-} \mathcal{H}(s, x, q, A^+, A^-) \geq 0.$$

(H8) There exists  $L_{H_3} > 0$  such that for all  $s, x, q \in G_H^\#, s', x', q' \in T \times X \times Q$  and  $A \in \mathbb{R}^d$ ,

$$|\mathcal{H}(s, x, q, A, A) - \mathcal{H}(s', x', q', A)| \leq L_{H_3}(|s - s'| + \|x - x'\| + |q - q'|).$$

Let  $\Phi : \Omega \rightarrow \mathbb{R}$ ,  $h = (\Delta x, \Delta p, \Delta t)$  and set

$$\begin{aligned} S_h^\Omega(x, q, p, s, \lambda; \Phi) = & \min \left( \lambda - \varphi_I, \mathcal{H}(s, x, q, D^+ \Phi(x, q, p, s), D^- \Phi(x, q, p, s)) + \right. \\ & \left. \frac{\lambda - \Phi(x, q, p, s)}{\Delta t} + \frac{\lambda - \Phi(x, q, p - \Delta p, s + \Delta t)}{\Delta p} \right). \end{aligned}$$

Now, consider the following scheme

$$S_h(x, q, p, s, \lambda; \Phi) = \begin{cases} S_h^\Omega(x, q, p, s, \lambda; \Phi) & \text{if } (x, q, p, s) \in \Omega \\ \lambda - \min_{w \in W(x, q), p' \geq \delta} \Phi(x, g(w, q), p', s) & \text{if } p = 0, \end{cases} \quad (5.50)$$

along with the following operator

$$\mathcal{F}(x, q, p, s, u, \nabla u) = \begin{cases} u - \varphi(x) \wedge \partial_s u + \partial_p u + H(s, x, q, \nabla_x u) & \text{if } (x, q, p, s) \in \Omega \\ u(x, q, p, s) - (Mu)(x, q, p, s) & \text{if } p = 0. \end{cases} \quad (5.51)$$

**Proposition 5.6.** *Let  $\Phi \in C_b^\infty(\Omega)$ . Under hypothesis (H6 – H8) and the CFL condition*

$$\Delta t \left( \frac{1}{\Delta p} + \frac{1}{\Delta x} \left\| \sum_{i=1}^d \partial_{A_i^+} \mathcal{H} + \partial_{A_i^-} \mathcal{H} \right\|_\infty \right) \leq 1 \quad (5.52)$$

*the discretization scheme (5.50) of (5.51) is stable, monotone and consistent.*

*Moreover, a solution  $u_h$  of (5.50) converges towards the solution  $u$  of (5.51) as  $h \rightarrow 0$ .*

*Proof.* The proof follows the lines used in the framework of Barles-Souganidis [10]. The goal is to show that the numerical scheme solutions' envelopes

$$\begin{aligned} \underline{u}(x', q', p', s') &= \liminf_{\substack{(x, q, p, s) \rightarrow (x', q', p', s') \\ h \rightarrow 0}} u_h(x, q, p, s) \\ \bar{u}(x', s', p', s') &= \limsup_{\substack{(x, q, p, s) \rightarrow (x', q', p', s') \\ h \rightarrow 0}} u_h(x, q, p, s), \end{aligned}$$

are respectively supersolution and subsolution of (5.51). Then, using the comparison principle in theorem 5.2, one obtains  $\bar{u} \leq \underline{u}$ . Since the inverse inequality is immediate (using the definition of limsup and liminf, one gets  $u \equiv \underline{u} = \bar{u}$  achieving thus the convergence.

Only the subsolution property of  $\bar{u}$  is presented next, the proof of the supersolution property of  $\underline{u}$  being very alike.

Firstly, the proofs shows that (5.50) is stable, monotone and consistent. Observe that  $S$  is proportional to  $-\Phi$  in the terms outside the Hamiltonian. Since fluxes  $\mathcal{H}$  are monotone by hypothesis (H7) (see [65] for details in monotone Hamiltonian fluxes) whenever the CFL condition (5.52) is satisfied, the monotonicity of  $S$  follows. The stability is ensured by the boundedness of  $\Phi$  and hypothesis (H6). Finally, hypothesis (H8) and lemma 5.2 are used in a straightforward fashion to obtain the consistency properties below:

$$\begin{aligned} \limsup_{\substack{(x',q',p',s') \rightarrow (x,q,p,s) \\ h \rightarrow 0}} S_h(x',q',p',s',\lambda;\Phi) &\leq \mathcal{F}^*(x,q,p,s,\Phi,\nabla\Phi) \\ \liminf_{\substack{(x',q',p',s') \rightarrow (x,q,p,s) \\ h \rightarrow 0}} S_h(x',q',p',s',\lambda;\Phi) &\geq \mathcal{F}_*(x,q,p,s,\Phi,\nabla\Phi) \end{aligned}$$

Now, choose  $\Phi \in C_b^\infty(\Omega)$  such that  $\bar{u} - \Phi$  has a strict local maximum at  $(x_0, q_0, p_0, s_0) \in \bar{\Omega}$  (without loss of generality assume  $(\bar{u} - \Phi)(x_0, q_0, p_0, s_0) = 0$ ).

First, suppose  $p_0 > 0$ . Then there exists a ball centered in  $(x_0, q_0, p_0, s_0)$  of radius  $r > 0$  such that  $\bar{u}(x, q, p, s) \leq \Phi(x, q, p, s)$ ,  $\forall (x, q, p, s) \in B((x_0, q_0, p_0, s_0), r) \subset \Omega$ . Construct sequences  $(x_\epsilon, q_\epsilon, p_\epsilon, s_\epsilon) \rightarrow (x_0, q_0, p_0, s_0)$  and  $h_\epsilon \rightarrow 0$  as  $\epsilon \rightarrow 0$  such that  $u_{h_\epsilon}(x_\epsilon, q_\epsilon, p_\epsilon, s_\epsilon) \rightarrow \bar{u}(x_0, q_0, p_0, s_0)$  and  $(x_\epsilon, q_\epsilon, p_\epsilon, s_\epsilon)$  is a maximum of  $u_{h_\epsilon} - \Phi$  in  $B((x_0, q_0, p_0, s_0), r)$ . Denote  $\zeta_\epsilon = (u_{h_\epsilon} - \Phi)(x_\epsilon, q_\epsilon, p_\epsilon, s_\epsilon)$ . (Remark that  $\zeta_\epsilon \rightarrow 0$  as  $\epsilon \rightarrow 0$ ).

Then,  $u_{h_\epsilon} \leq \Phi + \zeta_\epsilon$  inside the ball and since  $S(x_\epsilon, q_\epsilon, p_\epsilon, s_\epsilon, u_{h_\epsilon}(x_\epsilon, q_\epsilon, p_\epsilon, s_\epsilon); u_{h_\epsilon}) = 0$ , by the monotonicity property one obtains

$$S(x_\epsilon, q_\epsilon, p_\epsilon, s_\epsilon, \Phi(x_\epsilon, q_\epsilon, p_\epsilon, s_\epsilon) + \zeta_\epsilon; \Phi + \zeta_\epsilon) \leq 0.$$

Taking the limit (inf)  $\epsilon \rightarrow 0$  together with the consistency of the scheme, one obtains the desired inequality

$$\mathcal{F}_*(x_0, q_0, p_0, s_0, \Phi, \nabla\Phi) \leq 0.$$

Suppose now that  $p_0 = 0$ . Construct sequences  $(x_\epsilon, q_\epsilon, p_0, s_\epsilon) \rightarrow (x_0, q_0, p_0, s_0)$  and  $h_\epsilon \rightarrow 0$  as  $\epsilon \rightarrow 0$  such that  $u_{h_\epsilon}(x_\epsilon, q_\epsilon, p_0, s_\epsilon) \rightarrow \bar{u}(x_0, q_0, p_0, s_0)$ . Then,

$$\begin{aligned} \liminf_{\substack{x_\epsilon \rightarrow x_0 \\ q_\epsilon \rightarrow q_0 \\ s_\epsilon \rightarrow s_0 \\ h_\epsilon \rightarrow 0}} S_h(x_\epsilon, q_\epsilon, p_0, s_\epsilon, \Phi(x_\epsilon, q_\epsilon, p_0, s_\epsilon), \Phi) &= \liminf_{\substack{x_\epsilon \rightarrow x_0 \\ q_\epsilon \rightarrow q_0 \\ s_\epsilon \rightarrow s_0 \\ h_\epsilon \rightarrow 0}} (\Phi - M\Phi)(x_\epsilon, q_\epsilon, p_0, s_\epsilon) \\ &= (\Phi - (M\Phi)_*)(x_0, q_0, p_0, s_0). \end{aligned}$$

Since each  $u_{h_\epsilon}$  is a solution of (5.50), using lemma 5.2 the above expression yields at the point  $(x_0, q_0, p_0, s_0)$ :

$$\begin{aligned} 0 &= (\Phi - (M\Phi)_*)(x_0, q_0, p_0, s_0) \\ &\geq (\Phi - (M\Phi)^*)(x_0, q_0, p_0, s_0) \\ &\geq (\Phi - (M^+\Phi))(x_0, q_0, p_0, s_0) \\ &= \mathcal{F}_*(x_0, q_0, p_0, s_0, \Phi, \nabla\Phi) \end{aligned}$$

achieving the desired inequality.  $\square$

### 5.4.2 Numerical Simulations

For the numerical simulations, the numerical Hamiltonian  $\mathcal{H}$  is discretized using a monotone Local Lax-Friedrichs scheme [65] (where the two components of the gradient are explicit):

$$\begin{aligned} \mathcal{H}(t, x, q; a^+, a^-, b^+, b^-) &= H\left(t, x, q; \frac{a^+ + a^-}{2}, \frac{b^+ + b^-}{2}\right) - \\ &\quad c_a \left(\frac{a^+ - a^-}{2}\right) - c_b \left(\frac{b^+ - b^-}{2}\right) \end{aligned}$$

where  $a^\pm = D_i^\pm v$ ,  $b^\pm = D_j^\pm v$  and the constants  $c_a, c_b$  are defined as

$$c_a = \sup_{t, x, q, r} |\partial_{r_a} H(t, x, q, r)| \quad (5.53)$$

$$c_b = \sup_{t, x, q, r} |\partial_{r_b} H(t, x, q, r)|. \quad (5.54)$$

Setting  $u_h^\# = [{}^q v_{I_k}^n, {}^q v_{I^{1\pm k}}^n, \dots, {}^q v_{I^{d\pm k}}^n, {}^q v_{I_{k-1}}^n]$ , the equation

$$S_h(x_I, q, p_k, s_{n+1}, {}^q v_{I_k}^{n+1}; u_h^\#) = 0$$

allows an explicit expression of  $v^{n+1}$  as a function of past values  $v^n$ :

$${}^q v_{I_k}^{n+1} = \begin{cases} \varphi_I \vee {}^q v_{I_k}^n - \Delta t \left( \frac{{}^q v_{I_k}^n - {}^q v_{I_{k-1}}^n}{\Delta p} + \mathcal{H}(t_n, x_I, q, D^- {}^q v_{I_k}^n, D^+ {}^q v_{I_k}^n) \right) & \text{if } (I, q, k, n) \in \Omega^\# \\ \min_{w \in W(x_I, q), k' \geq \delta / \Delta p} g(w, q) {}^q v_{I_{k'}}^n & \text{if } k = 0. \end{cases} \quad (5.55)$$

In order to illustrate the work presented, a simple vehicle model is used in the simulations. This model allows an analytic evaluation of its autonomy and is suitable for an a posteriori verification of the results.

The switch dynamics is given by

$$g(w, q) = |q - w|.$$

The energetic dynamical model is given by  $f(u, q) = (-a_x + qu, -q(a_y + u))$ , where  $a_x, a_y > 0$  are constant depletion rates of the battery's electric energy and the reservoir's fuel (whenever the RE is on), respectively. The control domain is taken  $U = [0, u_{\max}]$ .

Considering this dynamics, an exact autonomy of the system can be evaluated analytically. Given initial conditions  $(x_0, y_0)$  the shortest time to empty the fuel reservoir is given by  $t^* = y_0 / (a_y + u_{\max})$ . The SOC evaluated at this instant is given by  $x(t^*) = x(0) - t^*(a_x - u_{\max})$ . If  $x(t^*) \leq 0$ , it means the fuel cannot be consumed fast enough before the battery is depleted. This condition can be expressed in terms of the parameters of the model as  $x_0(a_y + u_{\max}) \leq y_0(a_x - u_{\max})$ . In this case, the autonomy is given by  $T^0 = x_0 / (a_x - u_{\max})$ . Otherwise, the autonomy is given by  $T^1 = (x_0 + u_{\max} t^*) / a_x$ .

Simulations are running fixing  $\Delta p = 0.05$  and several  $\Delta x$ . The time step  $\Delta t$  is obtained using (5.52) to ensure the numerical stability. Different instances of  $x_0, y_0$  are tested, all with a lag of  $\delta = 1$ , and the theoretical and evaluated autonomy are compared. For numerical purposes, the set of initial energies is a ball of radius  $2\Delta x$  around  $(x_0, y_0)$  and the admissible region is set as  $K = [0, 1]^2$ . Remark that (6.13), (6.15) give a natural value  $\tilde{L}_K$  for the numerical boundary outside set  $K$ .

$(x_0, y_0)$	$\Delta x$	$\varepsilon$	Iterations	CPU <sup>1</sup> running time (s)
(0.5,0.5)	0.06	0.609	98	9.2
	0.05	0.090	99	13.3
	0.04	0.220	113	24.7
	0.03	0.049	128	51.9
	0.02	0.278	161	151.5
(0.3,0.8)	0.06	0.754	86	6.6
	0.05	0.102	84	8.5
	0.04	0.345	98	15.6
	0.03	0.041	108	31.5
	0.02	0.265	135	94.0

Table 5.1 – Convergence results and running times of an instance using  $a_x = 0.10$ ,  $a_y = 0.15$ ,  $u_{\max} = 0.07$ . The table indicates the difference  $\varepsilon = |T^* - s^*|$  from the theoretical autonomy as a function of the discretization step  $\Delta x$ . Two initial conditions are used. The theoretical autonomy for this instance is  $T^* = 6.5909s$  for  $(x_0, y_0) = (0.5, 0.5)$  and  $T^* = 5.5455s$  for  $(x_0, y_0) = (0.3, 0.8)$ .

The simulated instances use  $a_x = 0.10$ ,  $a_y = 0.15$ ,  $u_{\max} = 0.07$ . Tests are made using two initial conditions. Table 5.1 groups the error  $\varepsilon = |T^* - s^*|$  between exact autonomy and the autonomy (6.10) evaluated using the scheme (5.55) and the algorithm running times.

Figures 5.1, 5.2 and 5.3 show the minimum of the value function for all values of  $p \in P^\#$  for  $q = 0, 1$  and the corresponding reachable set of the instance  $(x_0, y_0) = (0.3, 0.8)$  at times  $s = 2.65$ ,  $s = 3.00$ ,  $s = 3.75$  respectively.

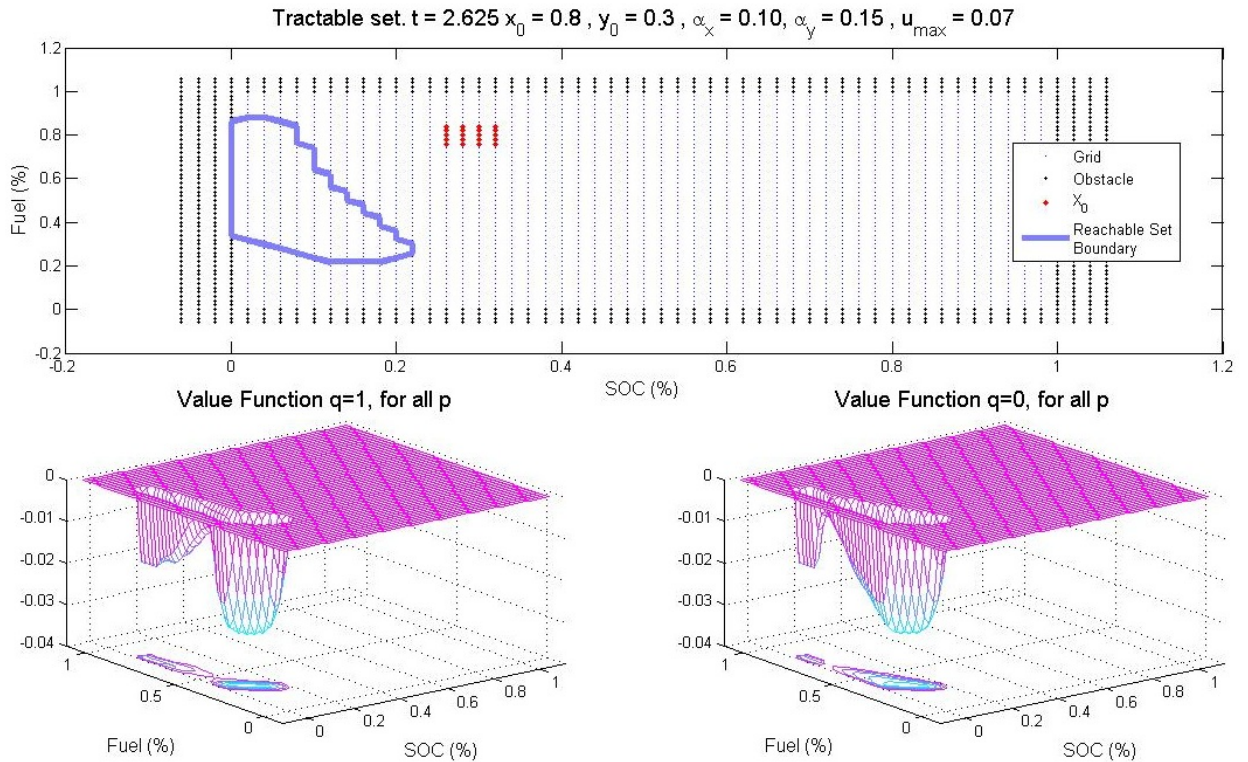
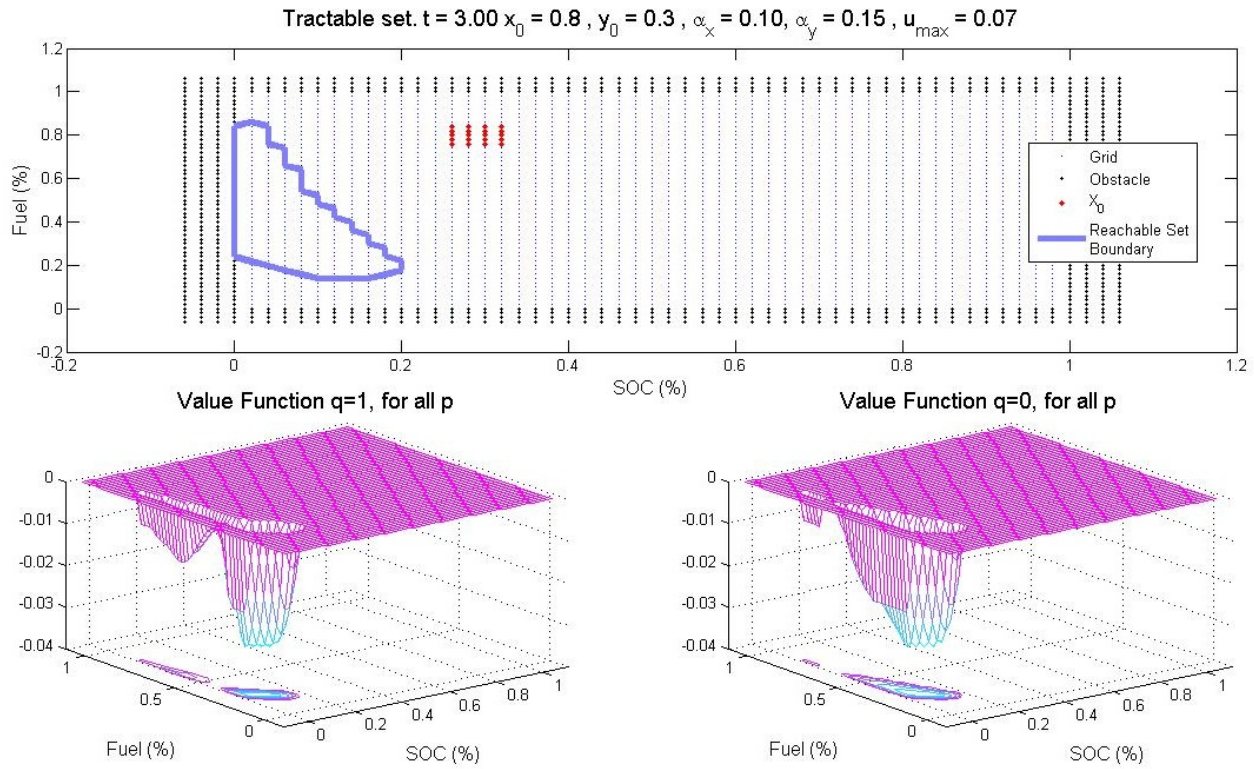
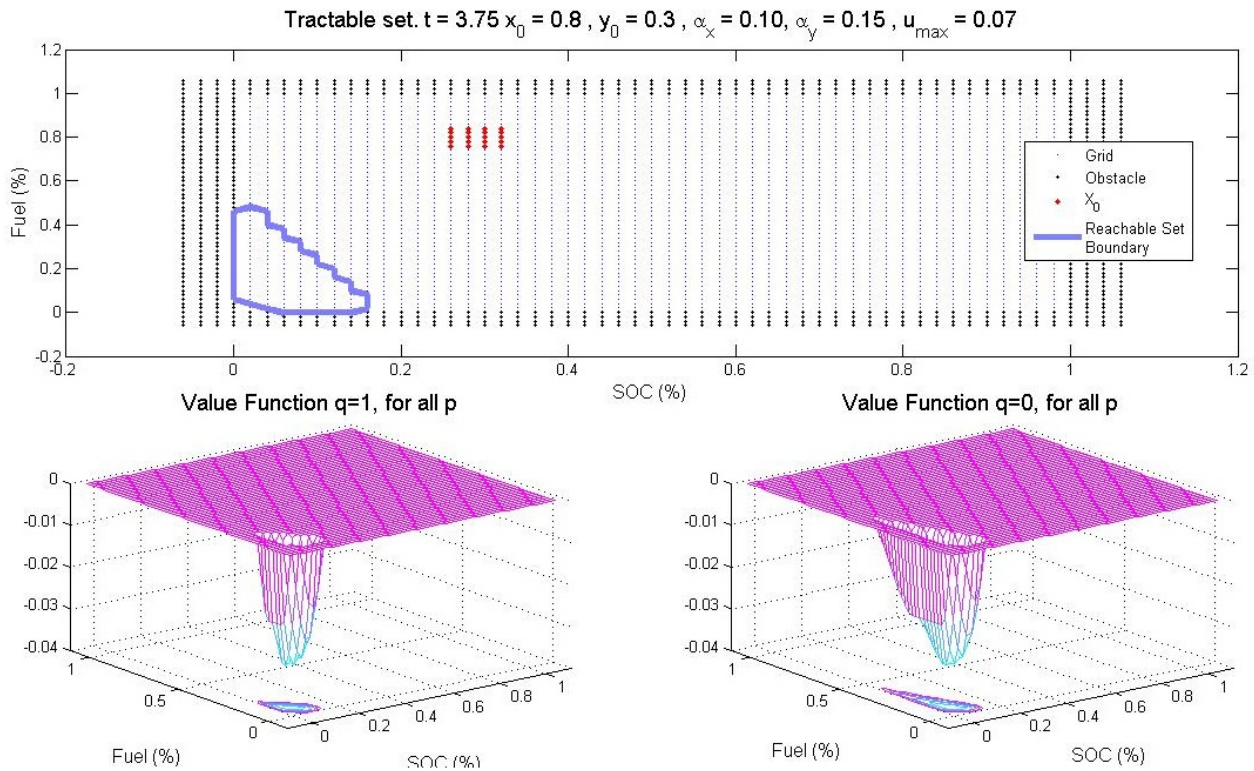


Figure 5.1 – Reachable set and value functions at  $s = 2.65$ .

From table 5.1 one can observe the convergence of the computed autonomy towards the theoretical value. This is a solid indication about the convergence of the numerical scheme towards the solution of the HJ equation. Even though this convergence is observed, one remarks that the CPU running time seems to grow in an exponential fashion as  $\Delta x$  decreases. This behavior is caused by

<sup>1</sup>Intel Xeon E5504 @  $2 \times 2.00\text{GHz}$ , 2.99Gb RAM.



Figure 5.2 – Reachable set and value functions at  $s = 3.00$ .Figure 5.3 – Reachable set and value functions at  $s = 3.75$ .

the conjunction of two main factors: the increase in the total mesh points considered in the scheme and the corresponding decrease in the time step needed to satisfy the CFL condition (5.52). In one hand, the increase in the number of mesh points naturally increases the computational overhead needed to evaluate the value function at each time step. In the other hand, the CFL constraints the time step  $\Delta t$  for a given discretization step  $\Delta x$ . In the instance considered in the numerical simulations, for step sizes of  $\Delta p = 0.15$  and  $\Delta x = \Delta y = 0.01$ , the time step must verify approximately  $\Delta t \leq 0.026$ . This time step requires roughly 300 iterations before reaching an autonomy of about 8s. With the perceived trend, the total running time is over 1600s. Although this is arguably still manageable in the simple dynamic system considered, when applied to a more realistic vehicle model the computation of the value function via the scheme (5.55) becomes prohibitive. Indeed, a more realistic simulation, even considering a relatively an initial state low on energy, such as  $(x_0, y_0) = (0.2, 0.2)$ , yields time autonomies of over 2400s ( $\approx 40\text{min}$ ). Treating such instance with a relatively coarser grid with a step size of  $\Delta x = 0.10$  would still require a time step of 1s or over 2000 iterations. With the supplementary computational overhead due the complexity of the vehicle model, the CPU time would be over a few hours, which is judged unreasonable for an application. This fact motivates the study of an alternative way of computing the value function developed in the next section.





# REACHABILITY OF HYBRID SYSTEMS: HYBRID VEHICLE APPLICATION

---

*“In theory there is no difference between theory and practice. In practice there is.”*

Albert Einstein

The content of this chapter is an adaptation from the following publications:

- Granato, G. *Reachability of Hybrid Systems Using Level-set Methods*, Optimization and Engineering, submitted, awaiting review.
- Granato G. *Range Optimization of Hybrid Vehicles*, Proceedings of the International Conference of Engineering Optimization - ENG OPT, 2012, Rio de Janeiro, Brazil, <http://www.engopt.org/paper/297.pdf>.

## Contents

---

6.1	Introduction . . . . .	133
6.2	Hybrid Vehicle Model . . . . .	134
6.3	Reachability Problem and Autonomy . . . . .	137
6.4	Optimal Control Problem and Dynamic Programming Principle . . . . .	138
6.5	Numerical Simulations . . . . .	143
6.6	Discussion . . . . .	149

---



## 6.1 Introduction

This section develops the study of the discrete-time dynamical system and the discrete-time optimal control problem for the reachability analysis of hybrid systems. In Chapter 5 theoretical aspects of a continuous-time system is studied. We show the the reachable set can be characterized by means of a level-set approach using the value function associated to the value function of an optimal control problem. This value function is shown to be the unique solution of a system of quasi-variational inequalities. Additionally, we show the convergence of a class of numerical schemes that allow the numerical computation of the value function, therefore enabling us to characterize the reachable set.

In this section, the focus on the discrete-time modeling of the hybrid system leads to, instead of a system of quasi-variational inequalities, a discrete-time dynamic programming principle which can be used to characterize the value function. More precisely, in this approach, the (discrete-time) reachable set is shown to be the level set of a value function associated to a optimal control problem. This value function is shown to satisfy a dynamic programming principle. Then, we construct an algorithm based on this dynamic programming principle to compute the value function. Once the value function is computed we characterize the reachable set and then use an associated minimum time function to synthesize a feedback control that controls the system to the desired energy state.

This study is motivated mainly by the hybrid vehicle application. The goal of this work is twofold. In one hand, given a geographic destination point and an associated route, if the energy on-board is sufficient to enable the vehicle to reach its destination, we can let the driver choose what will be the SOC and fuel at the destination point. In the other hand, if the vehicle is on a route that is too long respectively to the energy available on-board (i.e., not enough SOC – battery state of charge – nor fuel), one can compute the vehicle range. Since the discrete-time case of the broader study about the reachability analysis of hybrid systems is mainly motivated by the industrial application, the next sections are focused on the issues related to hybrid vehicles and the synthesis of feedback using the reachable set. This section seeks therefore to focus on the implementation of an algorithm to a vehicle application.

We recall briefly the main aspects of the range extender electric vehicle class of hybrid vehicles. According to [39], "a hybrid vehicle is a vehicle equipped with at least two energy sources for traction purposes". In this setting, the following question arises naturally: how much each power source shall contribute to meet the total power demand of the vehicle? The control policy of this power split is called power management strategy. In order to achieve such demanded power split between the energetic subsystems on-board the vehicle, the precise operation of the vehicle components varies according to the vehicle architecture and specifications. For instance, in the case of a hybrid series architecture containing an internal combustion engine (ICE) coupled to an alternator connected to the high-voltage network, the power split can be achieved by controlling the ICE torque and regime. Indeed, by controlling torque and regime, the total power output of the ICE is obtained, which in turn imposes the power output from the high-voltage battery. If at least two energy sources are present in the vehicle, it is clear that the vehicle maximum range becomes conditional to the control strategy. This paper addresses the issue of synthesizing the optimal power management strategy in the sense of maximizing the vehicle range.

As mentioned, this study focus on a range extender electric vehicle (REEV). This kind of hybrid vehicle consists of a primary power source - a high-voltage battery - and a small (powerwise) dimensioned ICE acting as an additional power source. Additionally, the REEV cannot rely solely on the ICE's power to drive the vehicle. In other words, the model considers that the vehicle's traction capability is conditioned to the existence of electric energy in the battery. The architecture is that of a series hybrid electric vehicle [73], which means that the ICE is not mechanically connected to the transmission. Instead, a generator transforms the mechanical energy produced in the ICE into electric current that can be directed towards the electric motor or charge the battery. Controls

available include the turning on and off of the ICE and the power produced by it. The controlled variables are the state of charge (SOC) of the battery, the remaining fuel on the ICE's reservoir and the ICE state on or off. In this setting, a power management strategy is a control sequence that dictates the state of the ICE - on or off - and if it is on, sets its output power. An important feature of the REEV model is a time interval  $\delta > 0$  imposed between two consecutive decisions times. From the physical viewpoint, this assumption incorporates the fact that frequent switching of the RE is undesirable in order to avoid mechanical wear off and acoustic nuisance for the driver.

This section is organized as follows: firstly, it describes the REEV model motivating the study and states the associated hybrid optimal control problem. Then, the reachable set and the value function are defined and a dynamic programming principle for the value function is obtained. It follows that the dynamic programming principle is used to derive an dynamic programming algorithm to compute the value function thus characterizing the reachable set of the system. Lastly, results of some numerical simulations evaluating the autonomy both of a REEV toy and realistic model are presented. We recall that our objective is to synthesize a controller that enables the driver to choose the SOC and fuel levels, among the possible alternatives, at the end of a given trip.

## 6.2 Hybrid Vehicle Model

This section discusses briefly (for a complete discussion, see Chapter 1) the model of the power management system of a REEV that is to be optimized and introduces the notation used throughout the document. A REEV is a vehicle that combines a primary power source - a HV battery - and a small dimensioned (powerwise) secondary power source - in our case an ICE. The traction (or propulsion) of the vehicle is performed by an electric motor connected to the vehicle's wheels through a reduction gear. Both energy sources can supply some share of the necessary power to meet the driver power request. Additionally, since the model considers a range-extender electric vehicle type, it cannot rely solely on the RE power to drive the vehicle. Thus, the model considers that the vehicle's traction capability is conditioned to the existence of some electric energy in the battery.

The ICE state is controlled by a discrete sequence of switching orders. An important feature of the REEV model is a time lag  $\delta > 0$  imposed between two consecutive decisions times. From the physical viewpoint, this assumption incorporates the fact that frequent switching of the RE is undesirable in order to avoid mechanical wear off and acoustic nuisance for the driver. This lag has an order of magnitude of about 120s.

The controlled variables are the state of charge (SOC) of the battery, the remaining fuel on the ICE reservoir and the ICE state on or off.

This setting considers that the optimal power management strategy is synthesized for a specific trip. In this direction, we assume that information about the vehicle's future power demand is known. In particular, it is assumed that the vehicle future speed, i.e. the speed profile of the vehicle, as well as the relief of the route followed by the vehicle are known. As a consequence, the problem can be stated in a deterministic setting. This may seem a very strong assumption. However, range extender electric vehicles are expected to be equipped with embedded navigation systems (NAV) that actually have information about expected speeds and topological data. This information can be exploited in order to construct an expected speed profile with a certain degree of confidence.

In the following of the paper, index  $k = 1, \dots, \kappa$  shall index a position in the route. Since the vehicle speed is assumed to be known, one can easily obtain, from a known location  $k$ , the time to attain it. More precisely, denote by  $d_k$  the distance between points  $k - 1$  and  $k$  and  $\sigma_k > 0$  the (constant) vehicle speed between points  $k - 1$  and  $k$ . The total time to drive between  $k - 1$  and  $k$  is then simply  $d_k/\sigma_k$ .

The state vector is composed of the vehicle energy state and the ICE state. The vehicle energy state is a two-dimensional vector  $y \in X = \mathbb{R}^2$ , where  $y = (y_1, y_2)$  denotes the state of charge of the battery and the fuel available in the range extender module respectively. Each of these quantities are the image of the remaining energy in the battery and in the RE respectively. Moreover, because energy capacities of the battery and the fuel tank are limited (and normalized between 1 – full reservoir – and 0 – empty reservoir), the set of admissible states is defined as  $K = [0, 1]^2$ . Remark that we define the state space as all the plane  $X = \mathbb{R}^2$  and introduce a set of admissible states  $K$ . The set  $K$  represents the "physically admissible" SOC and fuel levels, considering that they are normalized capacities. This is a modeling choice that later enables us to deal easily with the state constraints in the dynamic programming principle (in opposition to imposing "hard" constraints on the state space like  $X = [0, 1]^2$ ).

The RE state is denoted by  $q \in Q = \{0, 1\}$  and indicates whether the RE is off ( $q = 0$ ) or on ( $q = 1$ ).

The control variables are the output power of the ICE and the order to switch the ICE's state (on or off). The first is referred to as the continuous control and the latter as the discrete (or switch) control. The continuous control is applied at all points  $k = 0, \dots, \kappa - 1$  of the route and is denoted  $u_k \in U(q_k)$ .

The discrete control  $w$  is a sequence of switching decisions

$$w = \{(w_1, s_1), \dots, (w_N, s_N)\}, \quad (6.1)$$

where for all  $j = 1, \dots, N$ ,  $s_i \in \{1, \dots, \kappa - 1\}$  and  $w_j \in W(q_{s_j}) \subset Q$ . The sequence of switch controls  $\{w_j\}_{j=1}^N$  (designating the new mode of operation of the ICE) is associated with the sequence of switching positions  $\{s_j\}_{j=1}^N$  so that each decision  $w_j$  is exerted at node  $s_j$ . The set  $W(q)$  is a set value function of the actual discrete state and contains all possible new discrete states. In the vehicle application, where there are only two discrete modes of operation (on or off), this set is simply  $W(1) = \{0\}$  and  $W(0) = \{1\}$ . The switching decision is introduced so one is able to stop the range extender whenever there are large idle periods, thus saving fuel. More precisely, if the ICE is running and no output power is required, there is still an idle fuel consumption, as energy is required to keep the engine running to avoid stalling. In the case of extended idle periods, it may be worth to stop the ICE and pay a fuel overhead to restart the engine whenever needed.

The switch lag condition imposes that two consecutive switch decisions must be separated by a time interval of  $\delta > 0$ , i.e.

$$\sum_{i=s_j+1}^{s_{j+1}} \frac{d_i}{\sigma_i} \geq \delta.$$

Given a discrete control sequence  $w$  and a discrete state  $q_j$ , the continuous control steers the state according to

$$y_{k-1} = f_k(y_k, u_{k-1}, q_j)$$

and at isolated points  $\{s_j\}$ , the discrete state changes according to

$$q_{j-1} = g(q_j, w_j).$$

These expressions mean that: given a discrete control sequence  $w$ , let  $q_j$  be a mode of operation activated at some point  $s_{j+1}$ . Then, the dynamic  $f(\cdot, \cdot, q_j)$  operates on the state between points  $k = s_{j+1}, \dots, s_j$  and the trajectory of the discrete state is simply  $q_k = q_j$  for  $k$  between  $s_{j+1}$  and  $s_j$ . When point  $s_j$  is reached, the discrete dynamics undergoes a switch associated with the switch control  $w_j$  and changes to  $q_{j-1} = g(q_j, w_j)$ , activating the new mode of operation given by  $f(\cdot, \cdot, q_{j-1})$ . Observe moreover that the process is modeled and described in backward fashion.

We assume that functions  $f$  and  $g$  produce bounded increase, i.e., there exists  $L_f \geq 0$  such that, for all  $k = 0, \dots, \kappa$ ,  $y \in X$ ,  $q \in Q$  and  $u \in U(q)$ ,  $\|y - f_k(y, u, q)\| \leq L_f$  and there exists  $L_g \geq 0$  such that, for all  $q \in Q$  and  $w \in W(q)$ ,  $\|q - g(w, q)\| \leq L_g$ .

At this point, we introduce the notion of hybrid control and admissible hybrid control. The hybrid control is introduced for notational convenience. It is a control regrouping both the range extender power  $u$  and the switch decision  $w$ . More in detail, given a discrete control sequence  $w = \{(w_1, s_1), \dots, (w_N, s_N)\}$  and a continuous control sequence  $u = (u_1, \dots, u_{\kappa-1})$ , the hybrid control is denoted by  $a = (u, w)$  and groups the continuous and discrete controls. Denote by  $A$  the set of all hybrid controls. As mentioned, the hybrid control  $a$  must contain controls  $(u, w)$  such that  $u_k \in U(q_k)$  and  $w_j \in W(q_{s_j})$  for  $j = 1, \dots, N$ . However, the admissibility of each discrete and continuous control does not imply the admissibility of the hybrid control without additional structural conditions. By structural conditions we refer to the fact that the decision nodes  $s_j$  must be increasing with  $j$  and that the associated decision times must respect the lag condition. The next definition introduces the necessary structure for an admissible hybrid control.

**Definition 6.1.** A hybrid control  $a = (u, w)$  is said to be admissible if it has increasing decision nodes

$$s_1 \leq s_2 \leq \dots \leq s_N, \quad (6.2)$$

and satisfies the decision lag condition (where  $\delta > 0$ )

$$\sum_{i=s_j+1}^{s_{j+1}} \frac{d_i}{\sigma_i} \geq \delta. \quad (6.3)$$

Denote by  $A^{\text{adm}} \subset A$  the set of admissible hybrid controls.

The structure imposed on the admissible hybrid controls  $a \in A^{\text{adm}}$  allows the system to engender admissible trajectories. An admissible trajectory can be regarded as a "physically" admissible trajectory. Such trajectories correspond to the evolution of the SOC and fuel levels whenever the system is steered by an admissible control sequence. Fix  $\kappa > 0$ . Given a (not necessarily admissible) hybrid control  $a \in A$  and given final conditions  $x \in X$  and  $q \in Q$ , the hybrid dynamical system is written as

$$y_{k-1} = f_k(y_k, u_{k-1}, q_j), \quad k = \kappa, \dots, 1, \quad y_\kappa = x \quad (6.4)$$

$$q_{j-1} = g(q_j, w_j), \quad j = 1, \dots, N, \quad q_N = q. \quad (6.5)$$

Denote the solutions of (6.4)-(6.5) with final conditions  $x, q$  by  $y_{x,q;\kappa}$  and  $q_{x,q;\kappa}$ . As pointed out, not all discrete control sequences are admissible. Only admissible control sequences engender admissible trajectories. Thus, given  $\kappa > 0$ ,  $x \in X$  and  $q \in Q$ , the admissible trajectory set  $Y_{(0,\kappa)}^{x,q}$  is defined as

$$Y_{(0,\kappa)}^{x,q} = \left\{ y_{x,q;\kappa} \mid a \in A^{\text{adm}} \text{ and } y_{x,q;\kappa} \text{ solution of (6.4)-(6.5)} \right\} \quad (6.6)$$

Observe that the admissible trajectories set does not include the discrete trajectory.

The hybrid control admissibility condition formulated in conditions (6.2)-(6.3) is not well adapted to a dynamic programming principle formulation, needed later on. In order to include the admissibility condition in the optimal control problem in a more suitable form, we introduce a new state variable  $\pi$ . Recall that the decision lag condition implies that new switch orders are not available until a time  $\delta$  after the latest switch. The new variable is constructed such that, at a given position  $k$ , the value  $\pi_k$  measures the time since the last switch. The idea is to translate the structural conditions (6.2)-(6.3) into conditions satisfied by this new variable and treat them as state constraints. Thus, if  $\pi_k < \delta$  all switch decisions are blocked and if, conversely,  $\pi_k \geq \delta$  the

system is free to switch. For that reason, this variable can be seen as a switch lock. Now, given  $\kappa > 0$ ,  $k = 1, \dots, \kappa$  and a discrete control sequence  $w$ , the switch lock dynamics is defined by

$$\pi_k^w = \pi_k = \begin{cases} \delta + \sum_{i=1}^k \frac{d_i}{\sigma_i} & \text{if } k < s_1 \\ \inf_{s_j \leq k} \sum_{i=s_j}^k \frac{d_i}{\sigma_i} - \frac{d_{s_j}}{\sigma_{s_j}} & \text{if } k \geq s_1 \end{cases} \quad (6.7)$$

Indeed, once the discrete control is given, the trajectory  $\pi_k$  can be determined. Proceeding with the idea of adapting the admissibility condition we wish to consider  $\pi_\kappa = p$ , with  $p \in P := (0, T]$ , where  $T$  is the total travel time, the final value of the switch lock variable trajectory and impose the delay condition under the form  $\pi_{s_j-1} \geq \delta$  for all  $s_j$  (notice that  $\pi_{s_j} = 0$  by construction). These conditions suffice to define an admissible discrete control set. So, while optimizing with respect to admissible functions, one needs only look within the set of hybrid controls that engender a trajectory  $\pi_k$  with an appropriate structure. In other words, given  $\kappa > 0$ ,  $x \in X$ ,  $q \in Q$  and  $p \in P$ , define a admissible trajectory set  $S_{(0,\kappa)}^{x,q,p}$  as

$$\begin{aligned} S_{(0,\kappa)}^{x,q,p} &= \{y_{x,q,p;\kappa} \mid a = (u, \{w_j, s_j\}_{j=1}^N) \in A, y_{x,q,p;\kappa} \text{ solution of (6.4)-(6.5),} \\ &\quad \pi_{x,q,p;\kappa} \text{ solution of (6.7), } \pi_\kappa = p, \pi_{s_j-1} \geq \delta, j = 1, \dots, N\}. \end{aligned} \quad (6.8)$$

Following the above definitions, sets (6.6) and (6.8) satisfy

$$Y_{(0,\kappa)}^{x,q} = \bigcup_{p \in P} S_{(0,\kappa)}^{x,q,p}$$

In the following of the paper, whenever we wish to call attention to the fact that the final conditions of (6.4), (6.5) and (6.7) are fixed, we denote their solutions respectively by  $y_{x,q,p;\kappa}$ ,  $q_{x,q,p;\kappa}$ ,  $\pi_{x,q,p;\kappa}$  and their values at some point  $k$  by  $(y_{x,q,p;\kappa})_k$ ,  $(q_{x,q,p;\kappa})_k$ ,  $(\pi_{x,q,p;\kappa})_k$ .

### 6.3 Reachability Problem and Autonomy

Let  $X_0 \subset X$  be the set of allowed initial states, i.e. the set of states from which the system (6.4)-(6.5) is allowed to start. Define the reachable set as the set of all points attainable by  $y$  after driving until point  $k$  starting within the set of allowed initial states  $X_0$  to be

$$\begin{aligned} R_k^{X_0} &= \left\{ x \mid \exists (q, p) \in Q \times P, y_{x,q,p;k} \in S_{(0,k)}^{x,q,p}, \right. \\ &\quad \left. (y_{x,q,p;k})_0 \in X_0, \text{ and } (y_{x,q,p;k})_\theta \in K, \theta = 0, \dots, k \right\} \end{aligned} \quad (6.9)$$

In other words, the reachable set  $R_k^{X_0}$  contains the values of the final energy state  $(y_{x,q;k})_k$ , regardless of  $q, p$  at point  $k$ , for all admissible trajectories – i.e., trajectories obtained through an admissible hybrid control – starting within the set of possible initial states  $X_0$  that never leave set  $K$ .

In particular, the information contained in (6.9) allows one to determine the first point where the reachable set is empty. More precisely, given  $X_0 \subset X$ , define  $k^* \geq 0$  to be

$$k^* = \inf \{k \geq 0 \mid R_k^{X_0} \subset \emptyset\}. \quad (6.10)$$

The position (6.10) is identified as the *autonomy* of the hybrid system (6.4)-(6.5). Indeed, one can readily see that if no more admissible energy states are attainable after  $k^*$ , any admissible trajectory must leave set  $K$  of admissible values beyond this time. Therefore,  $k^*$  is interpreted as the last point in the path in which there is some usable energy on-board the vehicle.



In the case of range extender electric vehicles, the vehicle stops operating in nominal mode when the battery is depleted, even if there is some fuel left in the tank. If the driving conditions allows all the fuel to be consumed, then the last reachable energy state of the vehicle is the state  $(0, 0)$ , where there is no charge in the battery and no fuel in the tank. However, this is by all means the sole last reachable energetic state in all situations. Indeed, because the range extender is power-wise small dimensioned, it may be the case that the engine cannot consume energy quickly enough to empty the fuel tank. For instance, if the vehicle is driving through a steep slope, the battery may be depleted before all the fuel is consumed resulting thus in a final state  $(0, y_2)$ ,  $y_2 > 0$ .

The reachable set  $R_k^{X^0}$  is associated with the *minimum time function*  $\mathcal{T} : X \rightarrow \mathbb{R}^+ \cup \{\infty\}$ , defined as

$$\mathcal{T}(x) = \inf\{k \geq 0 \mid x \in R_k^{X^0}\} \quad (6.11)$$

The minimum time function associated to a point  $x$  in the state space is the minimum time that any admissible trajectory needs to reach  $x$  departing from set  $X_0$  while respecting the state constraints  $K$ . Observing the definition of autonomy (6.10), if the initial set  $X_0$  consists of only one point, the autonomy  $k^*$  is linked to the minimum time function by the relation

$$k^* = \sup\{\mathcal{T}(x) < \infty \mid x \in X\}. \quad (6.12)$$

In other words, the autonomy is the greatest (finite) minimum time of the system. This is not the case if  $X_0$  contains more than one point.

## 6.4 Optimal Control Problem and Dynamic Programming Principle

In order to characterize the reachable set  $R_k^{X^0}$  this paper follows the classic level-set approach [64]. The idea is to describe (6.9) as the negative region of a function  $v$ . It is well known that the function  $v$  can be defined as the value function of some optimal control problem. In the case of system (6.4)-(6.5),  $v$  happens to be the value function of a (discrete time) hybrid optimal control problem.

Consider a Lipschitz continuous function  $\tilde{\phi} : X \rightarrow \mathbb{R}$  such that

$$\tilde{\phi}(x) \leq 0 \Leftrightarrow x \in X^0.$$

Such a function always exists – for instance, the signed distance function  $d_{X_0}$  from the set  $X^0$ . For  $L_K > 0$ , one can construct a bounded function  $\phi : X \rightarrow \mathbb{R}$  as

$$\phi(x) = \max(\min(\tilde{\phi}(x), L_K), -L_K). \quad (6.13)$$

For a given point  $k \geq 0$  and hybrid state vector  $(x, q, p) \in X \times Q \times P$ , define the value function to be

$$v_k^0(x, q, p) = \inf_{S_{(0,k)}^{x,q,p}} \{\phi((y_{x,q,p;k})_0) \mid (y_{x,q,p;k})_\theta \in K, \forall \theta \in \{0, \dots, k\}\}. \quad (6.14)$$

Observe that (6.9) works as a level-set to the negative part of (6.14). Indeed, since (6.14) contains only admissible trajectories that remain in  $K$ , by (6.13) implies that  $v_k^0(x, q, p)$  is negative if and only if  $(y_{x,q,p;k})_0$  is inside  $X_0$ , which in turn implies that  $x \in R_k^{X^0}$ .

Now we proceed to describe the state constraints in the similar manner. The set  $X \setminus K$  of state constraints is treated as an *obstacle* in the state space, i.e., the state trajectory cannot pass through

the obstacle set  $K$ . We recall that  $K = [0, 1]^2$  is the normalized SOC and fuel quantities. More precisely, define a Lipschitz continuous function  $\tilde{\varphi} : X \rightarrow \mathbb{R}$  to be

$$\tilde{\varphi}(x) \leq 0 \Leftrightarrow x \in K,$$

and

$$\varphi(x) = \max(\min(\tilde{\varphi}(x), L_K), -L_K). \quad (6.15)$$

Then, for a given  $k \geq 0$  and  $(x, q, p) \in X \times Q \times P$ , define a total penalization function to be

$$J_k(x, q, p; y) = \left( \phi((y_{x,q,p;k})_0) \bigvee_{\theta \in \{0, \dots, k\}} \max \varphi((y_{x,q,p;k})_\theta) \right)$$

and then, the optimal value :

$$v_k(x, q, p) = \inf_{y \in S_{(0,k)}^{x,q,p}} J_k(x, q, p; y). \quad (6.16)$$

Here and in the rest of the paper, we denote  $a \vee b := \max(a, b)$ .

Observe that (6.14) and (6.16) are bounded thanks to the constructions (6.13) and (6.15) respectively. The idea in place is that one needs only to look at the sign of  $v_0$  or  $v$  to obtain information about the reachable set. Therefore, the bound  $L_K$  removes the necessity of dealing with an unbounded value function besides providing a convenient value for numerical computations. In order to ensure that constructions (6.13) and (6.15) do not interfere with the original problem's formulation (in the sense that using  $\phi, \varphi$  or  $\tilde{\phi}, \tilde{\varphi}$  should yield the same results), given  $k, X_0$ , it suffices to take  $L_K > \sup_{x \in X_0} \|x\| + kL_f$ , where  $L_f$  is the bound on  $|y_{k+1} - y_k|$  (so that the dynamics  $f$  will not allow the norm of the state to reach the value  $L_K$ ).

The next proposition certifies that (6.9) is indeed a level-set of (6.14) and (6.16).

**Proposition 6.1.** *Let  $X_0 \subset X$ . Define functions  $\phi$  et  $\varphi$  respectively by (6.15) and (6.13). Define value functions  $v^0$  et  $v$  respectively by (6.16) and (6.14). Then, for  $k > 0$ , the reachable set (6.9) is given by*

$$R_k^{X_0} = \{x \mid \exists (q, p) \in Q \times P, v_k(x, q, p) \leq 0\} = \left\{x \mid \exists (q, p) \in Q \times P, v_k^0(x, q, p) \leq 0\right\}. \quad (6.17)$$

Proposition (6.1) sets the equivalence between (6.9) and the negative regions of (6.14) and (6.16). In particular, it states that it suffices to compute  $v$  or  $v_0$  in order to obtain information about  $R^{X_0}$ . In this sense, this paper focuses on (6.16), which is associated with an optimal control problem with no state constraints.

The value function can also be used to characterize the minimum time function. In addition to the minimum function, the next proposition introduces the *extended minimum time function*, which takes into account all state variables  $x, q, p$ . This extended function is needed later to the controller synthesis.

**Proposition 6.2.** *The minimum time function  $\mathcal{T} : X \rightarrow \mathbb{R}^+ \cup \{\infty\}$  is given by*

$$\mathcal{T}(x) = \inf \{k \mid \exists (q, p) \in Q \times P, v_k(x, q, p) \leq 0\}. \quad (6.18)$$

*The extended minimum time function  $\mathcal{T}' : X \times Q \times P \rightarrow \mathbb{R}^+ \cup \{\infty\}$  is given by*

$$\mathcal{T}'(x, q, p) = \inf \{k \mid v_k(x, q, p) \leq 0\}. \quad (6.19)$$

Observe that equation (6.18) is an immediate consequence of the level set identities in proposition 6.1. Relation (6.19) characterizing the extended minimum time function using the value function is derived in a similar fashion.

Before stating a dynamic programming principle satisfied by (6.16), we introduce some preliminary notations.

Given  $\kappa > 0$ , set  $\mathcal{K} = \{1, \dots, \kappa\}$ ,  $\Omega = X \times Q \times P \times \mathcal{K}$  and denote its closure by  $\bar{\Omega}$ . (We recall that, concerning the vehicle application,  $X = \mathbb{R}^2$ ,  $Q = \{0, 1\}$ ,  $P = (0, T]$ , where  $T$  is taken to be the total travel time). As such, (6.16) is defined as

$$\begin{aligned} v : \bar{\Omega} &\rightarrow \mathbb{R} \\ (x, q, p, k) &\mapsto v_k(x, q, p) \end{aligned}$$

Define

$$\mathcal{V}(\bar{\Omega}) := \{v \mid v : \bar{\Omega} \rightarrow \mathbb{R}, v \text{ bounded}\}.$$

Now, define the non-local switch operator  $M : \mathcal{V}(\bar{\Omega}) \rightarrow \mathcal{V}(\bar{\Omega})$  to be (adapted from the ideas in [90])

$$(Mv_k)(x, q, p) = \inf_{\substack{w \in W(q) \\ p' \geq \delta}} v_k(x, g(w, q), p')$$

The action of this operator on the value function represents a switch that respects the delay constraint. Thus, they only make sense when  $p = 0$ .

The next proposition states the dynamic programming principle verified by (6.16).

**Proposition 6.3.** *The value function (6.16) satisfies the following dynamic programming principle:*

(i) For  $k=0$ ,

$$v_0(x, q, p) = \phi(x) \bigvee \varphi(x), \quad \forall (x, q, p) \in X \times Q \times P, \quad (6.20)$$

(ii) For  $p = 0$ ,

$$v_k(x, q, 0) = (Mv_k)(x, q, 0), \quad \forall (x, q, k) \in X \times Q \times \mathcal{K}, \quad (6.21)$$

(iii) Otherwise,

$$v_k(x, q, p) = \inf_{u_{k-1}} \left\{ v_{k-1} \left( f_k(x, u_{k-1}, q), q, p - \frac{d_k}{\sigma_k} \right) \bigvee \varphi(x) \right\}. \quad (6.22)$$

*Proof.* The dynamic programming principle is composed of three parts.

(i): Equality (6.20) is obtained directly from definition (6.16).

(ii): " $\leq$ ". Consider hybrid controls  $a = (u, \{w_j, s_i\}_{i=1}^N)$  and  $\bar{a} = (\bar{u}, \{\bar{w}_j, \bar{s}_j\}_{j=1}^{N-1})$  such that  $\bar{u} = u$ ,  $\bar{w}_j = w_j$ ,  $\bar{s}_j = s_j$  for  $j = 1, \dots, N-1$  and  $w_N = w'$ ,  $s_N = k$ . For short, denote by  $y \equiv y_{x,q,0;k}$  the trajectory associated with control  $a$  and  $\bar{y} \equiv y_{x,g(q,w'),p';k}$  the trajectory associated with control  $\bar{a}$ . Then, it follows that

$$\begin{aligned} v_k(x, q, 0) &\leq J_k(x, q, 0; y) \\ &= J_k(x, g(q, w'), p', \bar{y}). \end{aligned}$$

In order to the switch  $(w', k)$  be admissible,  $p' \geq \delta$ . Since  $\bar{a}$  is arbitrary, one can choose it such that

$$\begin{aligned} v_k(x, q, 0) &\leq \inf_{\bar{y} \in S_{(0,k)}^{x,g(q,w'),p'}} J_k(x, g(q, w'), p', \bar{y}) + \epsilon_1 \\ &= v_k(x, g(q, w'), p') + \epsilon_1, \end{aligned}$$

where  $\epsilon_1 > 0$ . Now, choose the last switch  $w'$  and  $p'$  such that

$$\begin{aligned} v_k(x, q, 0) &\leq \inf_{\substack{w' \in W(q) \\ p' \geq \delta}} v_k(x, g(q, w'), p') + \epsilon_1 \\ &= (Mv_k)(x, g(q, w'), p') + \epsilon_1. \end{aligned}$$

"  $\geq$ ". Choosing an  $\epsilon_2$ -optimal controller  $a_\epsilon$  and denoting by  $y^\epsilon$  the associated trajectory, one has

$$v_k(x, q, 0) + \epsilon_2 \geq J_k(x, q, 0; y^\epsilon).$$

Proceeding with the same hybrid control constructions as in the " $\leq$ " case, one obtains

$$\begin{aligned} J_k(x, q, 0; y^\epsilon) &= J_k(x, g(w', q), p', s; \bar{y}^\epsilon) \\ &\geq v_k(x, g(w', q), p') \\ &\geq \inf_{\substack{w' \in W(q) \\ p' \geq \delta}} v_k(x, g(w', q), p') \\ &= (Mv_k)(x, q, 0). \end{aligned}$$

Relation (6.21) is then obtained by the arbitrariness of both  $\epsilon_1, \epsilon_2$ .

(iii): Consider a hybrid control  $a = (u, w)$  where  $u = (u_0, \dots, u_{k-1})$ . From (6.16) one obtains

$$\begin{aligned} v_k(x, q, p) &= \inf_{a \in A} \phi((y_{x,q,p;k})_0) \bigvee_{\theta \in \{0, \dots, k\}} \varphi((y_{x,q,p;k})_\theta) \\ &= \inf_{a \in A} \max \left( \left( \phi((y_{x,q,p;k})_0) \bigvee_{\theta \in \{0, \dots, k-1\}} \varphi((y_{x,q,p;k})_\theta) \right), \varphi((y_{x,q,p;k})_k) \right) \\ &= \inf_{a \in A} \max \left( \left( \phi((y_{x,q,p;k})_0) \bigvee_{\theta \in \{0, \dots, k-1\}} \varphi((y_{x,q,p;k})_\theta) \right), \varphi(x) \right). \end{aligned} \quad (6.23)$$

The next relations allows one to rewrite the trajectory  $y_{x,q,p;k}$  in terms of its value at point  $k-1$  instead of  $k$ :

$$(y_{x,q,p;k})_0 = (y_{(y_{x,q,p;k})_{k-1}, q, p-d_k/\sigma_k; k-1})_0 \quad (6.24)$$

$$\max_{\theta \in \{0, \dots, k-1\}} \varphi((y_{x,q,p;k})_\theta) = \max_{\theta \in \{0, \dots, k-1\}} \varphi((y_{(y_{x,q,p;k})_{k-1}, q, p-d_k/\sigma_k; k-1})_\theta) \quad (6.25)$$

Now, construct hybrid controls  $a^0 = ((u_0, \dots, u_{k-2}), w)$  and  $a^1 = ((u_{k-1}), w)$ . Since  $\pi_k = p > 0$  the discrete control sequence  $w$  does not switch at the point  $k$ . (We recall that, following (6.7), a switch at point  $s$  is characterized by  $\pi_s = 0$ ). Therefore, hybrid control  $a^0$  corresponds to the hybrid control sequence  $a$  up to point  $k-1$  and  $a^1$  corresponds to the hybrid control at point  $k$ , namely,  $u_{k-1}$ .

The above conclusion together with relations (6.24), (6.25) allow (6.23) to be rewritten as

$$v_k(x, q, p) = \inf_{u_{k-1}} \left\{ v_{k-1} \left( (y_{x,q,p;k})_{k-1}, q, p - \frac{d_k}{\sigma_k} \right) \bigvee \varphi(x) \right\}. \quad (6.26)$$

Finally, equation (6.22) is obtained by plugging (6.4) in (6.26).

□

The dynamic programming principle allows one to compute the value function and consequently characterize the reachable set  $R_k^{X_0}$ . Once the value function is obtained, one can synthesize a

controller using the minimum time function  $\mathcal{T}$ . The computation of the value function and the subsequent controller synthesis are obtained by a dynamic programming algorithm. The algorithm is the projection of the dynamic programming principle (cf. proposition 6.3) over a grid  $\mathcal{X} \times Q \times \mathbf{P}$  of the discretized state space.

**Algorithm 6.1** (Dynamic Programming Algorithm). *Step 1: Compute value function.*

1.1 Given route information  $\sigma_j, d_j$ ,  $j = 1, \dots, \kappa$  (recommended speeds and link lengths respectively), set  $k = 1$  and construct a discretization of the state space  $\mathcal{X}, \mathbf{P}$ ;

1.2 For all  $(x, q, p) \in \mathcal{X} \times Q \times \mathbf{P}$  set  $v_k(x, q, p) = \phi(x) \vee \varphi(x)$ ;

1.3 For  $k = 2, \dots, \kappa$ , for all  $(x, q, p) \in \mathcal{X} \times Q \times \mathbf{P}$  compute

$$\begin{aligned} v_k(x, q, p) &= (Mv_k)(x, q, p), \text{ if } p = 0, \\ v_k(x, q, p) &= \inf_{u_{k-1}} \left\{ v_{k-1}^\# \left( f_k(x, u_{k-1}, q), q, p - \frac{d_k}{\sigma_k} \right) \vee \varphi^\#(x) \right\} \text{ otherwise.} \end{aligned}$$

Once having evaluated  $v$  in  $\mathcal{X} \times Q \times \mathbf{P}$  at all times the controller can be synthesized as follows:

*Step 2: Control synthesis.*

2.1 Set  $k \in \{1, \dots, \kappa\}$ , and select reachable final state  $(x_k, q_k, p_k) = (x', q', p')$  such that  $v_k(x', q', p') \leq 0$ ;

2.2 Compute switch control: If  $p_k = 0$  set  $w_k^* = 1$  and update

$$\begin{aligned} q_k &\leftarrow g(w_k, q_k), \\ p_k &\leftarrow \arg \min_{p \geq \delta} \mathcal{T}'^\#(x_k, q_k, p). \end{aligned}$$

Otherwise, set  $w_k^* = 0$  and update

$$p_k \leftarrow p_k - \frac{d_k}{\sigma_k} \wedge 0.$$

Compute power control:

$$u_k^* = \arg \min_u \mathcal{T}'^\#(f_k(x_k, u, q_k), q_k, p_k)$$

2.3 Update state to  $x_k \leftarrow f_k(x_k, u_k^*, q_k)$ , set  $k \leftarrow k - 1$ . If  $k = 1$  terminate the algorithm, otherwise go to 2.2.

Here,  $v_{k-1}^\#, \varphi^\#, \mathcal{T}'^\#$  denote the interpolate of the value function, obstacle function and extended minimum time function at a point in the grid  $\mathcal{X} \times Q \times \mathbf{P}$  of the discretized state space, respectively. In step 2.1, one chooses  $k$  to be the node located at the destination if it is reachable. If this is not the case,  $k$  may be set to  $k^*$  in order to obtain the controller that maximizes the vehicle range.

Algorithm 6.1 outputs a controller  $(u^*, w^*)$  that controls the range extender power output and its mode of operation to reach the destination point after a time  $\kappa$  such that the final state is  $(x', q', p')$  departing from a state in  $X_0$ . Additionally, the trajectory remains inside set  $K$ , which in our case amounts to respect the fuel and SOC capacities. One can observe that the extended minimum time function plays a major role in the synthesis of the controller. Indeed, the value function numerical values have no physical meaning since the obstacle and target functions  $\phi$  and  $\varphi$  are arbitrary. Only the sign of the value function is needed to determine the reachable set. Instead, the physical information is encoded in the minimum time function. In the next section, we present numerical simulations illustrating algorithm (6.1).

## 6.5 Numerical Simulations

Proposition 6.3 provides the expressions that allow the computation through a dynamic programming algorithm. Once the value function obtained, one can use (6.17) to characterize the reachable set and (6.10) to obtain the autonomy of the system.

First, a simple vehicle model is used in the simulations. In addition, a constant vehicle speed is assumed during this first simulation phase. This model allows an analytic evaluation of the autonomy and is suitable for an a posteriori verification of the results. After the algorithm is tested using this simple configuration, we introduce a realistic vehicle model and a realistic speed profile.

In this simple setting, we assume that the vehicle follows a constant speed of  $\sigma \equiv 1\text{m/s}$  and the distance between any two points  $k-1$  and  $k$  is  $d_k = 1\text{m}$ . We consider that  $X = \mathbb{R}^2$  and  $Q = \{0, 1\}$ . The switch dynamics are given simply by

$$g(w, q) = w,$$

ensuring that  $w$  imposes the new mode of operation at every switching time.

The continuous state dynamic models a simple REEV charge behavior. The evolution of the energetic state  $(x, y) \in \mathbb{R}^2$  – respectively, the battery SOC and the range extender fuel – is given by

$$f(u, q) = \begin{pmatrix} -a_x + qu \\ (-a_y - u)q \end{pmatrix}, \quad (6.27)$$

where  $a_x, a_y > 0$  are constant depletion rates of the battery electric energy and the reservoir fuel (whenever the RE is on), respectively. The coefficient  $a_x$  models a non-negative power requirement throughout the driving cycle and therefore, a non-negative SOC consumption. However, the net SOC variation can be made zero or even positive (i.e., charge the battery) by the action of the output power of the range extender  $u$ . To do so, the range extender must be running and fuel must be consumed. The factor  $a_y$  represents the fuel consumed to generate the necessary power to counter the inertia of moving parts of the range extender while  $u$  is the effective share of power transferred to the electric motor. In order to simplify the analysis we consider the conversion factors from units of power to units of (normalized) fuel mass and percentage of battery SOC to be unitary.

The engine maximum power is denoted by  $u_{\max}$  and the minimum power is set to be 0.

We start by assuming that the vehicle has enough available energy to reach its final geographic destination. Thus, one application of this study is to compute the reachable set at the destination and by proceeding as such, to enable the driver to chose among any SOC and fuel level at the destination. After the driver chooses a SOC and fuel level among the possible choices, the algorithm synthesizes a controller that will drive the range extender so to reach the desired energetic state.

Considering dynamics (6.27), an exact autonomy of the system can be evaluated analytically. Since there are no non-linearities in the model, an optimal strategy to reach the furthest point is to empty completely the fuel as soon as possible, thus obtaining the maximum amount of energy available. Given initial conditions  $(x_0, y_0)$  the shortest time to empty the fuel reservoir is given by

$$t^* = \frac{y_0}{a_y + u_{\max}}.$$

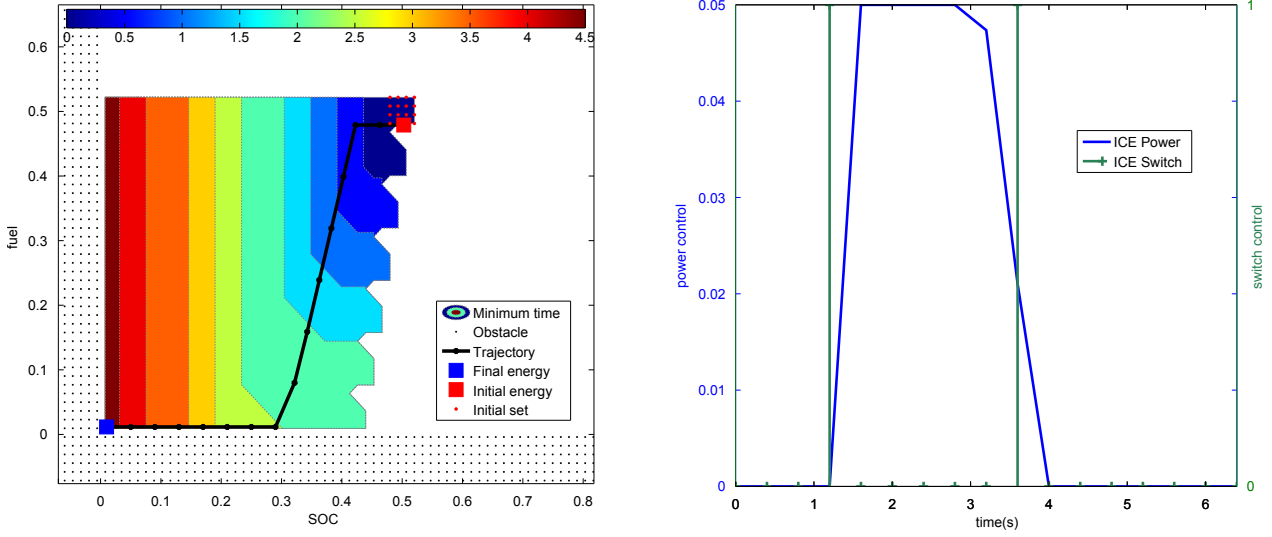
The SOC evaluated at this instant is given by  $x(t^*) = x(0) - t^*(a_x - u_{\max})$ . If  $x(t^*) \leq 0$ , it means the fuel cannot be consumed fast enough before the battery is depleted. This condition can be expressed in terms of the parameters of the model as  $x_0(a_y + u_{\max}) \leq y_0(a_x - u_{\max})$ . In this case, the autonomy is given by

$$T^0 = \frac{x_0}{a_x - u_{\max}}. \quad (6.28)$$

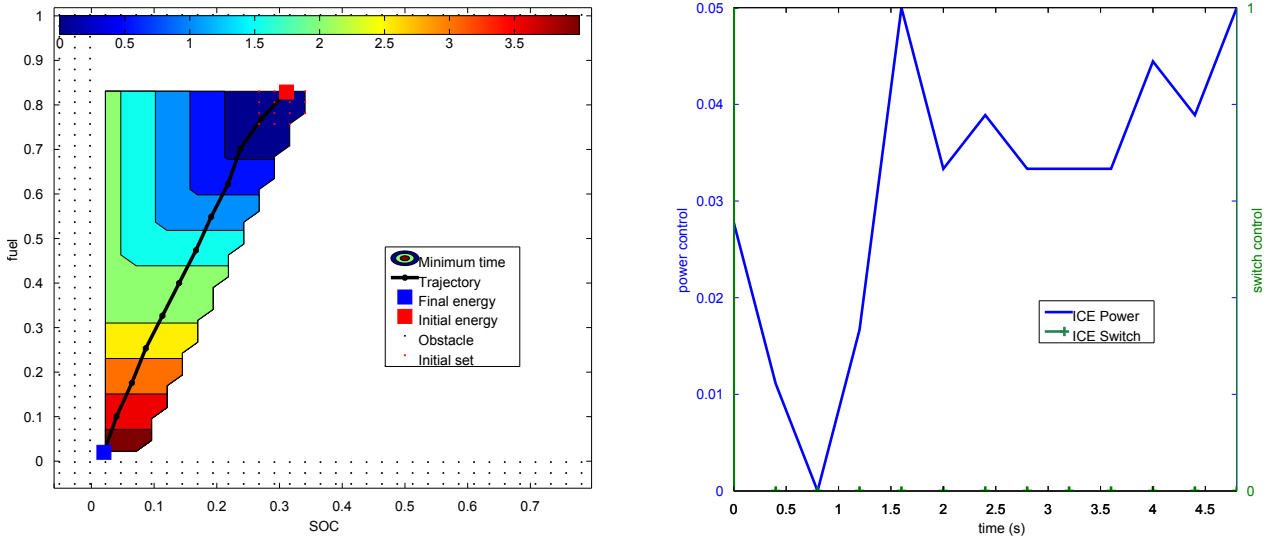
Autonomy (6.28) is the time needed to spend all the energy on board, using the range extender at full power.

In the other hand, if  $x(t^*) > 0$ , all the fuel in the range extender can be consumed and there is still some time until the battery is emptied. In this case, the autonomy of the system given by

$$T^1 = \frac{x_0 + u_{\max} t^*}{a_x}. \quad (6.29)$$



(a)  $(x_0, y_0) = (0.5, 0.5)$



(b)  $(x_0, y_0) = (0.3, 0.8)$

Figure 6.1 – Figure showing the trajectory of the energy state from a valid initial state until the last reachable point. The plot on the left groups the contour levels of the minimum time function, the obstacle and initial sets. On the right, the figure displays the range extender power control  $u$  and the switch sequence  $w$  controlling the ICE state. The synthesis is obtained using step sizes of  $\Delta x = 0.0245$ ,  $\Delta p = 0.4211$ ,  $\Delta t = 0.4$ .

Figure 6.1 illustrates controllers synthesized using algorithm 6.1 for two initial conditions. They present the contour plots of the minimum time function over the reachable states. One can readily see that the reachable set, and in particular, the trajectory remains inside the authorized region of the state space  $K = [0, 1]^2$ . Also, the controllers respect the delay condition  $\delta = 1$  between two consecutive switches. The initial set is taken to be a ball around  $(x_0, y_0)$  in each case for numerical reasons. Indeed, when considering only one point, the interpolation procedure for the value function, obstacle function and extended minimum function becomes a delicate matter.

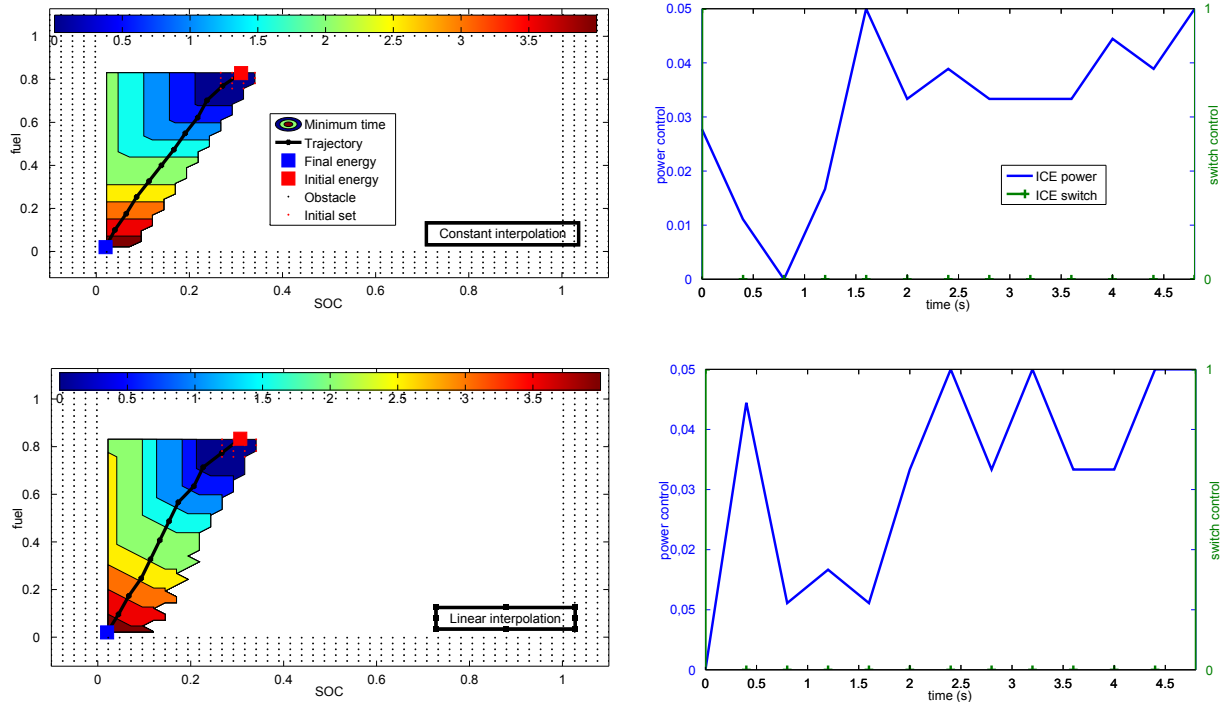


Figure 6.2 – Figure illustrating the differences in the controllers when using a constant and linear interpolation of  $v_{k-1}^\#, \varphi^\#, \mathcal{T}^\#$  in algorithm 6.1

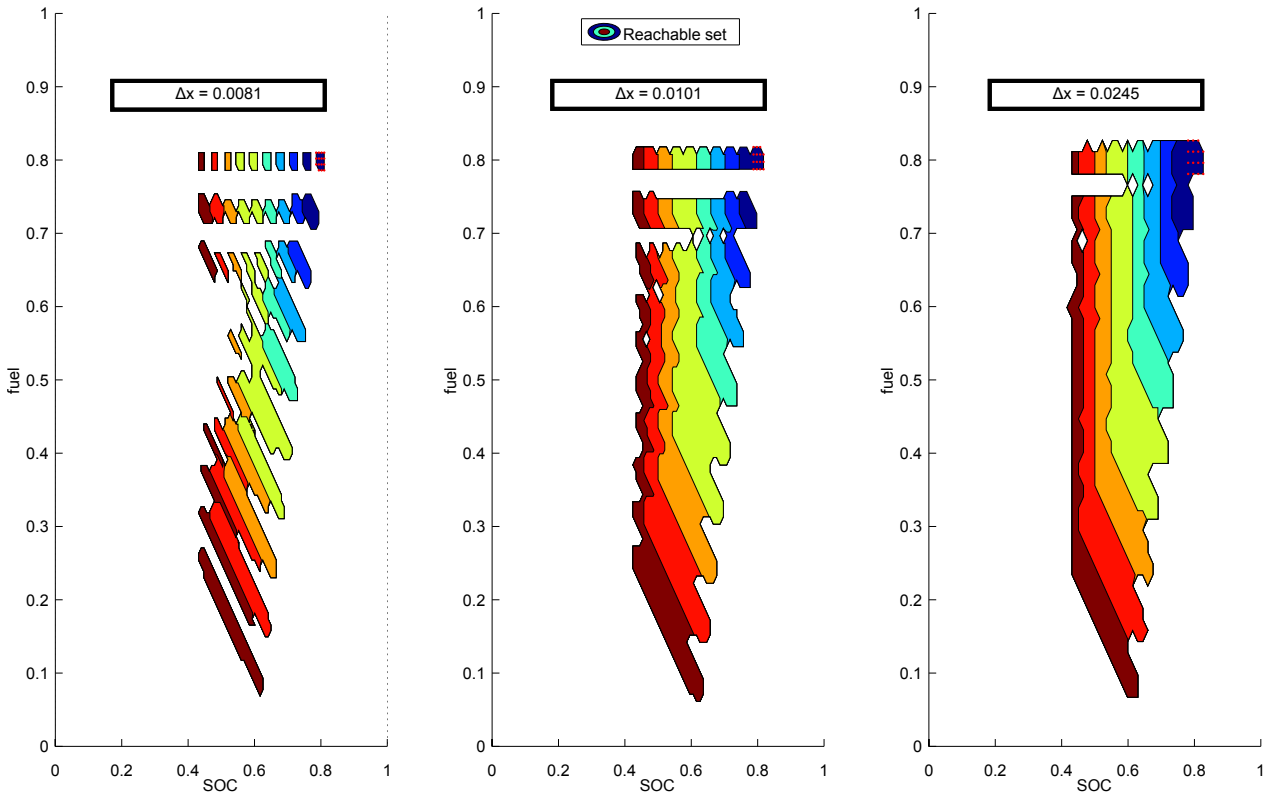


Figure 6.3 – Figure showing the dissipative effect occurring when using a linear interpolation of function  $v_{k-1}^\#, \varphi^\#, \mathcal{T}^\#$  appearing as the energy grid step size decreases. The instances consider  $\Delta x = 0.0081$ ,  $\Delta x = 0.0101$  and  $\Delta x = 0.0152$  respectively.



$(x_0, y_0)$	$\Delta x$	$\varepsilon$	CPU <sup>1</sup> running time(s)
(0.5,0.5)	0.05	3.291	2.48
	0.04	1.154	2.89
	0.03	0.218	4.21
	0.02	0.081	8.55
	0.01	0.054	19.01
(0.3,0.8)	0.05	1.536	1.91
	0.04	0.600	2.36
	0.03	0.336	3.34
	0.02	0.072	6.95
	0.01	0.191	16.75

Table 6.1 – *Convergence results and running times. Fixed  $\Delta t = 0.4$  and  $\Delta p = 0.5$ . Instance considering  $a_x = 0.10$ ,  $a_y = 0.15$ ,  $u_{max} = 0.07$*

Table 6.1 presents the error between the theoretical and computed autonomy as well as the CPU time required for two initial conditions in the instance using  $a_x = 0.10$ ,  $a_y = 0.15$ ,  $u_{max} = 0.07$  and  $\delta = 1$ . One can observe the convergence of the algorithm towards the theoretical value as  $\Delta x$  approaches 0. A main advantage of this algorithm is that the convergence is observed even holding the time step  $\Delta t$  fixed. Empirically, a negligible sensibility with respect to  $\Delta p$  is observed.

The interpolation method of functions  $v_{k-1}^\#, \varphi^\#, \mathcal{T}^\#$  plays a role in the synthesis of the controller. Figure 6.2 presents two interpolation methods used in the numerical simulations, namely, a constant and a linear interpolation. The CPU time required for the controller synthesis is of 21.34s when using a constant interpolation and of 24.89s when using linear interpolation. These interpolation methods ensure that the monotonicity of the functions are preserved (i.e., their suprema and infima points). The constant interpolation is less precise than its linear counterpart. However, when using a linear interpolation method, we observe that for smaller grid sizes  $\Delta x$  a dissipative effect on the propagation of the reachable set appears. Figure 6.3 illustrates this effect. We remark that the results illustrated in figure 6.1 are obtained using a constant interpolation method. However, the gain in precision obtained using the linear interpolation is judged to outweigh the larger CPU time and thus, is the method select to be used in the controller synthesis for the vehicle model. Nevertheless, one must keep in mind that the dissipation observed has to be taken into account when creating the discretization grid of the state space.

These remarks about the behavior of the algorithm allow us to proceed to the study of the reachable set in the realistic vehicle model. The procedure for implementing the simulations is as follows. We consider a driving profile consisting of  $\kappa > 0$  nodes and  $\kappa - 1$  links. At each node  $k$ , entry to the link  $k$  there is an associated link speed  $\sigma_k$  and the length of the link  $d_k$ . Figure 6.4 shows the speed profile  $\sigma_k$  and the length of the some links  $d_k$ . This cycle is a 19.6km cycle obtained from the recommended speed profile of a navigation system for a test track. Once  $\sigma_k, d_k$  are known, one can fix the time steps  $\Delta t_k$  used in the algorithm. Notice that the time steps depend on the link, the problem being non-autonomous in this case.

We proceed to the computation of the autonomy using a realistic vehicle model. Table 6.2 summarizes the main results. Figure 6.5 regroups the minimum time function, the synthesized controller that allows the vehicle to reach its maximum range and the corresponding evolution of the SOC and fuel. The simulations consider two initial conditions, namely, (0.3, 0.3) and (0.6, 0.4). The CPU time required is, respectively, of 921s and 1658s. The computed autonomy is respectively of  $T_{(0.3,0.3)}^* = 3195s$  and  $T_{(0.6,0.4)}^* = 5400s$ . This is equivalent to a range of  $\text{Range}_{(0.3,0.3)}^{\max} = 45.126km$  and  $\text{Range}_{(0.6,0.4)}^{\max} = 78.405km$ . The autonomies computed in both cases can be compared to the autonomy of the vehicle under a purely electric utilization. In the first case, the utilization of

<sup>1</sup>Intel Xeon E5504 @  $2 \times 2.00GHz$ , 2.99Gb RAM.

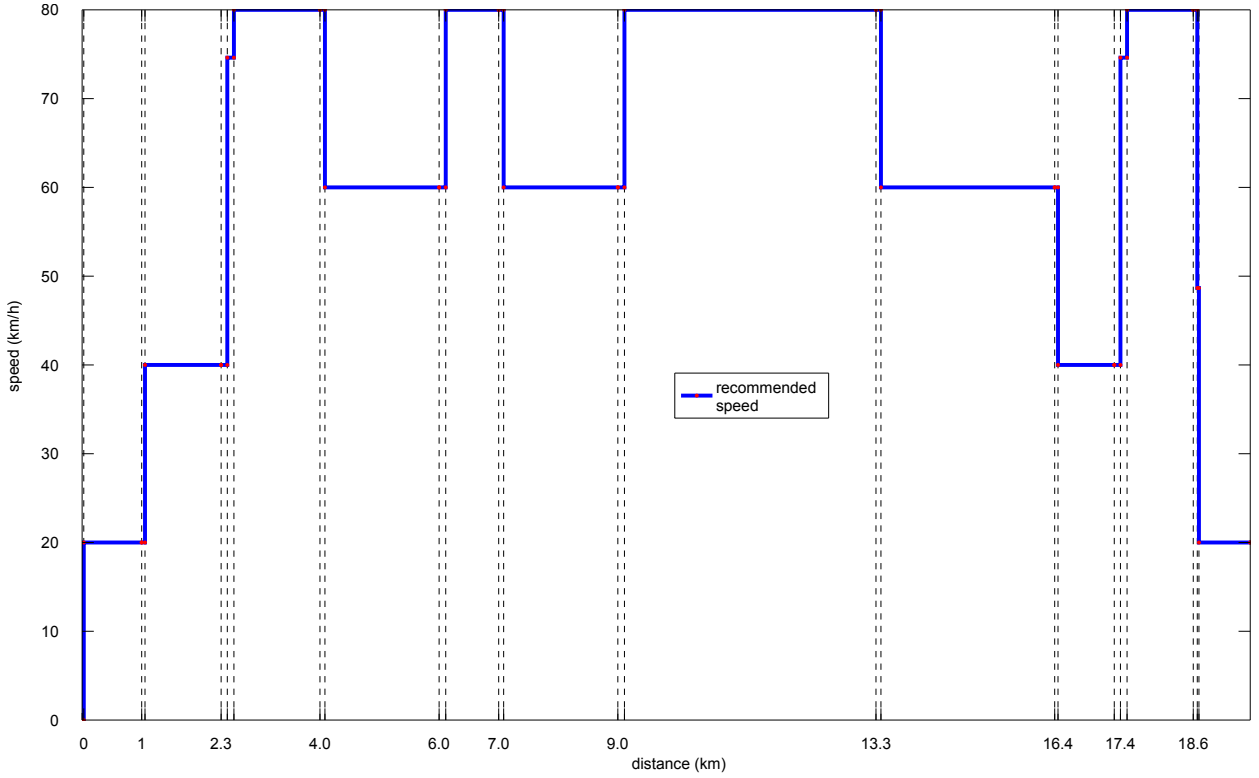


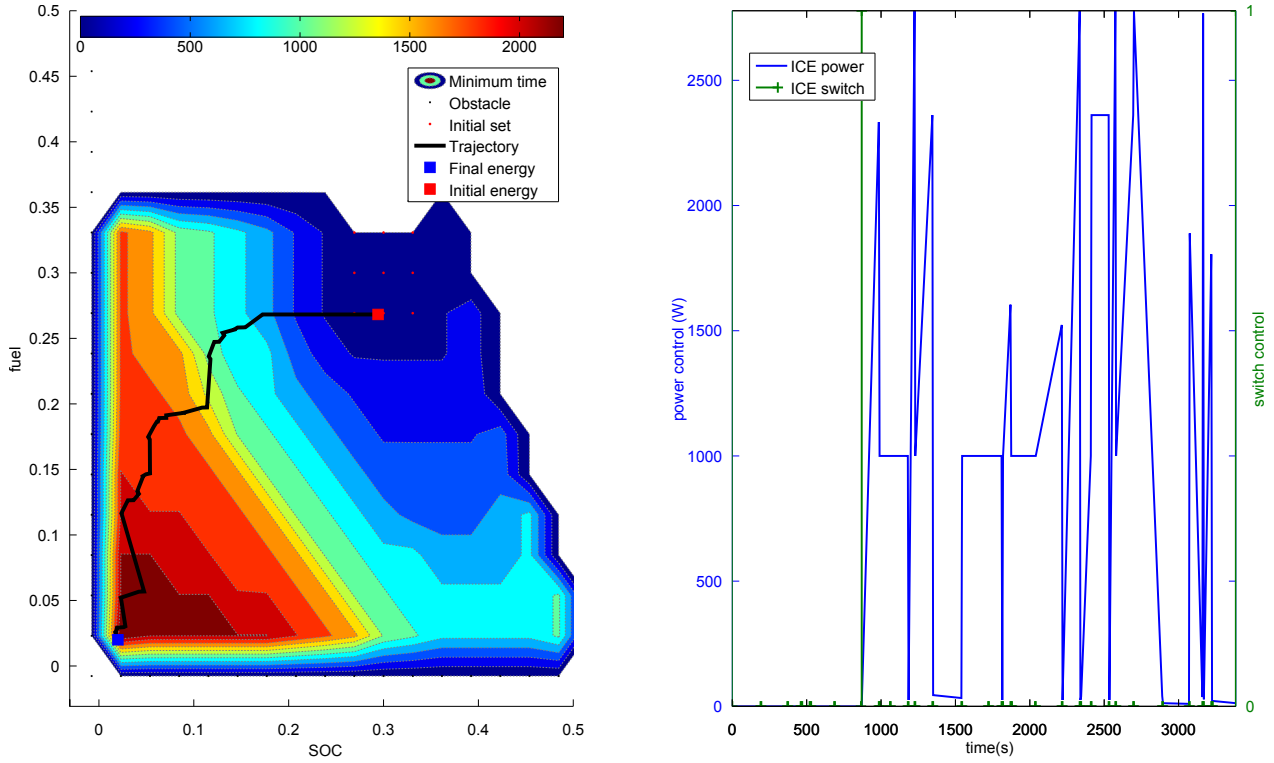
Figure 6.4 – Speed profile used in the simulations of vehicle model. The speed is extracted from the recommended speed profile of a navigation system of a 19.6km driving cycle. The dashed lines indicate the discretization points of the route on which the control decision is taken. Each red dot (and the corresponding dashed line) indicate the entry nodes of each link.

the range extender, when operated following the controller synthesized by algorithm 6.1, roughly doubles the range of the vehicle (a 104.7% range increase). When considering the initial condition with a SOC at 60% and fuel at 40%, the range extender increases the vehicle range by over 60%. The increase in range relatively to the purely electric utilization due the optimal utilization of the range extender (in the sense that the maximum range is reached) depends on the relative levels of initial SOC and fuel. We also consider the total operating *financial cost* (given in euros in this study) of the range extender. This cost is based on an estimated  $\eta = 1.5 \text{ €/l}$  of fuel. The range extender operating cost is then normalized with respect to 100km to yield a more meaningful result. Denoting the reservoir capacity by  $\bar{y}$ , the operating cost is calculated as

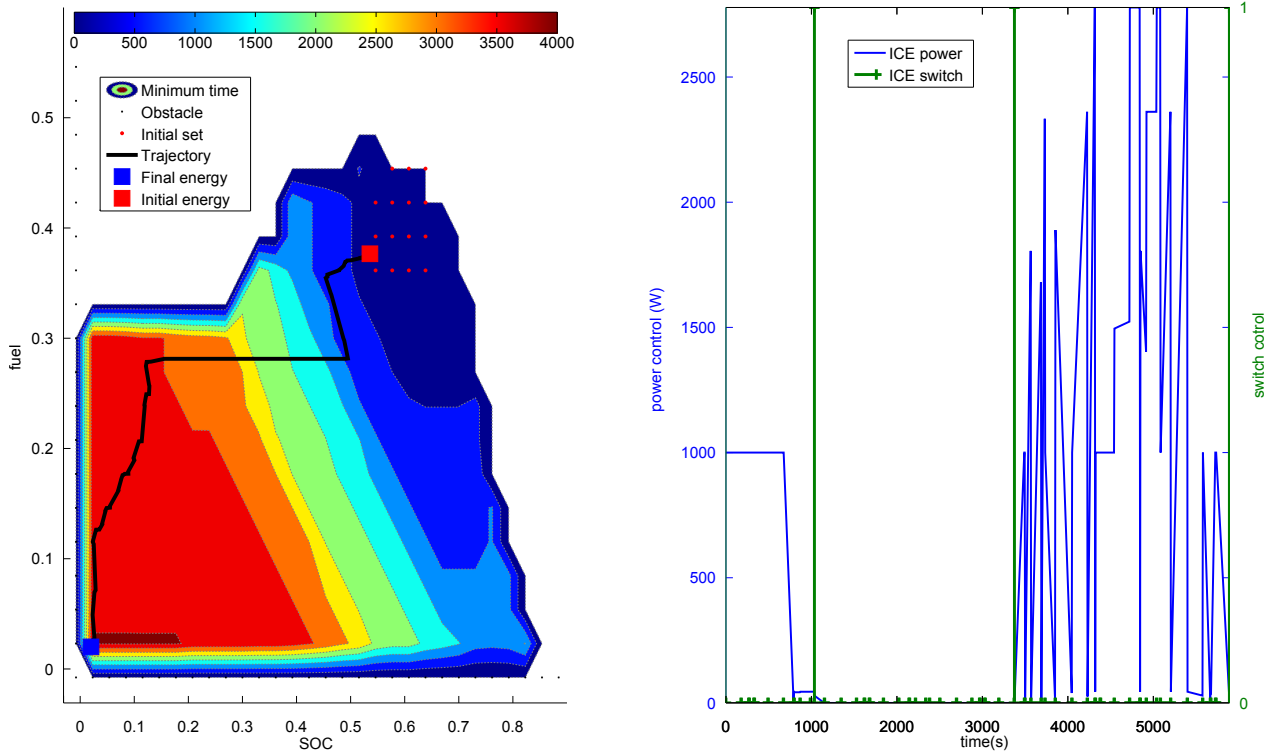
$$\text{Op. cost} = 100\eta\bar{y} \left( \frac{y_0}{\text{Max range} - \text{EV range}} \right). \quad (6.30)$$

We stress that we take into consideration in the cost of the range extender operation only the *additional distance* due the range extender. Therefore, the values of operating cost should be read as euros per 100km of range extender utilization instead of 100km of driving distance. In our model the reservoir has a total capacity of  $\bar{y} = 6\text{l}$ . From table 6.2, one can observe that the total cost of operation is stable with respect to the two initial conditions considered.

In addition to the vehicle range optimization, the algorithm also allows the driver to choose a final SOC/fuel repartition among the reachable states. More precisely, as the driver enters his/her final destination in the navigation system and the navigation system proposes a route to the destination point, the first step of algorithm 6.1 evaluates the reachable set at the destination. The reachable set is then presented to the driver where the selection of a desired final energy distribution is possible. The algorithm then synthesizes the controller based on the driver choice. Finally, the range extender is operated accordingly, thus reaching the desired final state. This versatile application is illustrated in Figure 6.6. The figure represents 3 different control strategies, each of which steers the vehicle energy state to one of the preselected final energy distributions. Departing from state (0.6, 0.4),



(a) Initial condition at  $(x_0, y_0) = (0.3, 0.3)$ . Computed autonomy  $T_{(0.3,0.3)}^* = 3195\text{s}$ . Vehicle range  $\text{Range}_{(0.3,0.3)}^{\max} = 45.126\text{km}$ .



(b) Initial condition at  $(x_0, y_0) = (0.6, 0.4)$ . Computed autonomy  $T_{(0.6,0.4)}^* = 5400\text{s}$ . Vehicle range  $\text{Range}_{(0.6,0.4)}^{\max} = 78.405\text{km}$ .

Figure 6.5 – Figure showing the trajectory of the energy state from an valid initial state until the last reachable point. The plot on the left groups the contour levels of the minimum time function, the obstacle and initial sets. On the right, the figure displays the range extender power control  $u$  and the switch sequence  $w$  controlling the ICE state. The synthesis is obtained using step sizes of  $\Delta x = 0.0308$ . We consider 30 grid points in the  $p$  dimension in both cases.

Initial SOC, fuel	(0.3, 0.3)	(0.6, 0.4)
Max range	45.126 km	78.405 km
EV range	22.045 km	48.209 km
Relative range increase	104.70%	62.64%
Range extender op. cost	11.70 €/100km	11.92 €/100km

Table 6.2 – *Results for the autonomy of the range extender electric vehicle. The table presents the maximum range of the vehicle when equipped with a range extender (Max range) and the range of the same vehicle when operating in a purely electric mode (EV range). This allows us to evaluate the relative range increase due the utilization of the range extender, operated following the controller synthesized by algorithm 6.1. The financial cost of operation of the range extender is estimated, based on a price of 1.5 €/l.*

the selected final states represents three classes of choices: a high final SOC/low final fuel selection representing a privileged range extender utilization; a low final SOC/high final fuel state representing an economic/ecologic range extender operation and a balanced final state. In the right side of Figure 6.6, the control strategy that steers the energy state to the driver selection is shown. Notice that the final states selection does not correspond to the same final point, as the time to reach each of the final states is different.

One difficulty of the numerical implementation of the algorithm relates to the step size  $\Delta p$  (and more generally to the curse of dimensionality). The domain of  $p \in [0, P]$  must include the possibility of a strategy that never switches. This can be the case only if  $P$  is greater than the vehicle autonomy. More precisely, the switch lock variable may be seen as a countdown time towards a switch. Denoting  $T^*$  the autonomy of the vehicle, if  $P < T^*$ , the longest one can wait before switching is  $P$ . This means that for an initial value of  $p = P$ , the ICE is obliged to switch after a time  $P$ , which occurs before the end of the trip. Therefore, if  $P < T^*$ , a strategy that never switches can never be synthesized. As any realistic application must make room for such a strategy, the domain of the switch lock variable must be  $P \geq T^*$ . That being said, the typical expected autonomy of a vehicle with 30% of SOC and fuel at a mean speed of 50km/h is about 60km which yield of a time 1.2 hours. The lag time is of  $\delta = 120$ s. A step size of 120s for an autonomy of 1.2 hours yields 70 grid points in the  $p$  dimension. Together with a energy state discretized with  $\Delta x = \Delta y = 0.2$  (or 50 grid points) and 70 time steps (about one each minute), the total number of points in the state space is  $50 \times 50 \times 70 \times 70 \times 2 = 24.5 \times 10^6$  points. Using double precision to store each value, the storage of the value function only takes about 200Mb. Ergo, even for this conservative instance, 70 grid points in the  $p$  dimension reveals itself prohibitively expensive in terms of physical memory required. As a consequence, in order to reach a more manageable problem size, one has to accept a compromise between the versatility of strategies and the quality of the solutions. Figure 6.2 shows the behavior of the reachable set when a zero-switch strategy is not accepted. Notice that after approximatively 20min (1200s, near the point (0.4, 0.4)) there is no strategy that allows one to save fuel by keeping the range extender off. This strategy is admissible, but the algorithm cannot take it into account due the above remarks.

## 6.6 Discussion

In this study we present an algorithm to determine the autonomy of a hybrid vehicle. The algorithm uses information about the driving profile of the vehicle, obtained typically through a navigation system. Once the destination is fixed and if there is not enough energy on-board the vehicle to reach the destination point, the algorithms computes the longest driving distance and synthesize the associated control sequence of the controllable power source. Otherwise, if the final destination is within reach, given the total available energy on-board, the algorithm enables the driver to choose the final energy distribution between the energy sources among the set of possible energy distributions. In our application, we focus on the range extender electric vehicle-class of

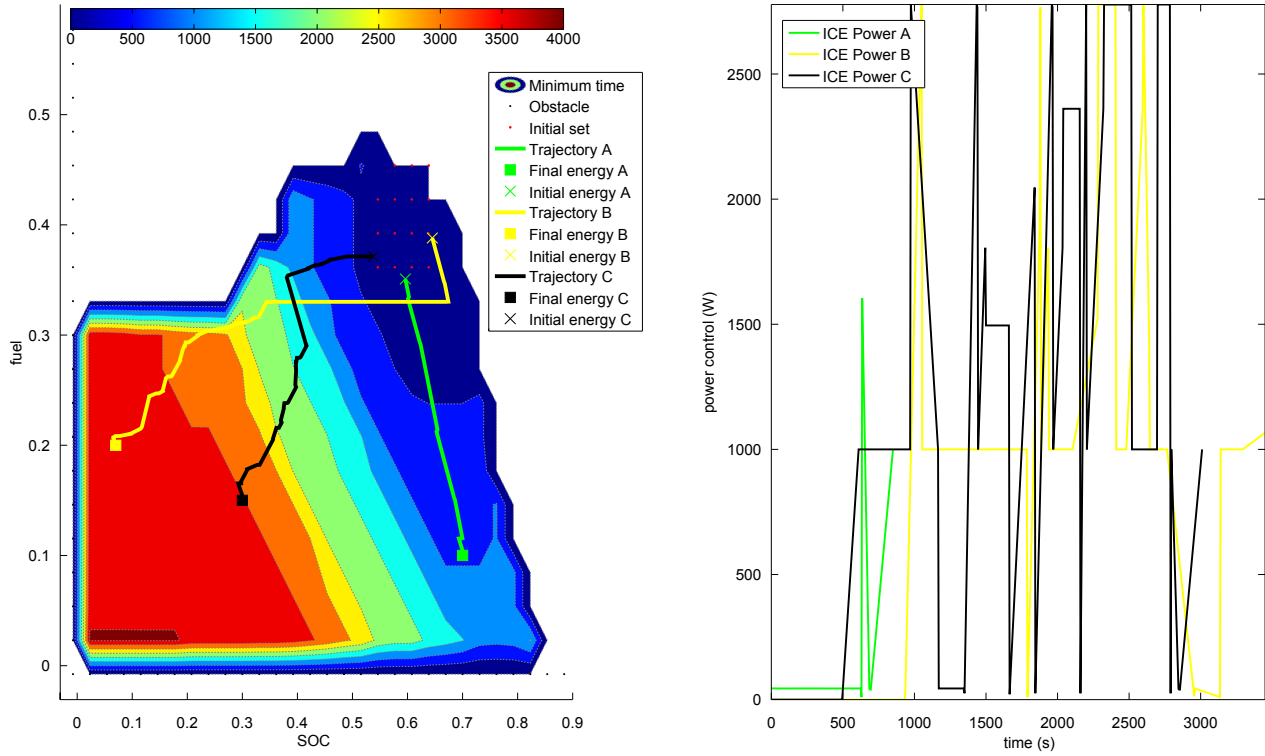


Figure 6.6 – Figure illustrating examples of trajectories where one selects the final energy distribution of the vehicle. The initial energy distribution of the vehicle considered is (around) the state  $(x_0, y_0) = (0.60, 0.40)$ . The final states chosen are  $(0.70, 0.10)$ ,  $(0.07, 0.20)$  and  $(0.30, 0.25)$ , corresponding respectively to trajectories A, B, C. They represent, respectively, points with high SOC/low fuel, low SOC/high fuel and a balanced final distribution, relatively to the initial state.

hybrid vehicles. Our model considers an internal combustion engine as a range extender and a high voltage DC battery as main energy source. The dynamic evolution of the state is considered to be a discrete-time evolution.

The algorithm computes the reachable set, set of reachable states, given a final time and initial state, via a level-set approach. The reachable set is characterized as the negative region of a value function associated to an optimal control problem. To compute the value function, we formulate an optimal control problem with obstacle and we state a dynamic programming principle verified by the optimal control problem. Then, the value function can be computed using a (deterministic) dynamic programming principle.

Numerical simulations of this procedure are performed first using a hybrid vehicle toy model and then using a realistic range extender electric vehicle model. The results shows that, depending on the initial conditions, operating the range extender as described in this paper may yield a 100% increase in the vehicle range, with respect to a purely electric vehicle. Moreover, numeric implementation aspects are discussed. Finally, controllers that operate the range extender to reach a driver-chosen final SOC/fuel distribution are presented.

# STOCHASTIC MODELING OF SPEED PROFILES

---

*“What we observe is not nature itself, but nature exposed to our method of questioning.”*

Werner Heisenberg

## Contents

---

7.1	Introduction . . . . .	153
7.2	Data Set Analysis and Model Selection . . . . .	155
7.3	Consumption Forecast Simulations . . . . .	158
7.4	Markov Chain Transition Probability Estimation . . . . .	160

---



## 7.1 Introduction

The core of this document concerns the study of optimal control strategies for the energy management of hybrid vehicles. A share of this investigation deals with stochastic control strategies. This means that the strategies take into account some uncertainty related to the model. In our case, the uncertainty models the vehicle future power demand. The random future power demand is considered a function of the uncertain vehicle future speed and other known variables. The goal of this section is twofold: first, to construct a model for the vehicle future speed, seen as a stochastic process. Second, to apply the stochastic model in forecasting the energy consumption of an electric vehicle, enabling therefore an estimation of its autonomy.

This Chapter details the construction of a stochastic model of the speed profile of an electric vehicle. The model can then be used to describe the power demand of the vehicle in order to obtain an estimation of the vehicle energy consumption. Additionally, the construction of a stochastic model for the vehicle speed is an important step in the construction of an optimization model needed later for the synthesis and simulation of stochastic optimal power management strategies.

*Energy consumption* is understood as the consumption of whatever form of energy is used to meet the power requirements of the vehicle. Aiming at predicting the autonomy of the vehicle, the central idea is to construct a model that enables us to forecast the power demand of the vehicle at some point of the route (in the future) given a set of available information (in the present). The integration of this power demand gives an estimation of the energy needs of the vehicle which in turn is used to obtain the vehicle estimated autonomy. In the context of electric vehicles this task is considered critical because of their relatively low range (between 80km and 200km).

Energy consumption estimation is based on two components: a *consumption model* and a *power demand profile*. A consumption model is a model, generally a look-up table, linking a set of variables to the *instant* energy consumption. Works [3, 62] construct such models by linking vehicle drive-like variables such as speed and acceleration directly to fuel consumption. Also, other exogenous factors are also linked to the fuel consumption, ranging from the vehicle model and type of the route to driver age and weather conditions [35]. A statistical analysis of a set of measurements is used to describe the relations between such variables (or group of variables) and energy consumption and to obtain a consumption model. A method for the construction of a stochastic driving cycle is proposed in [57]. They consider a Markov process of *model events*, i.e. regime phases (idle, cruise, acceleration, deceleration) of the driving profile of the vehicle. The transition matrix for this Markov process is then identified using data from California Air Resources Board, same data used to creating the Unified LA92 driving cycle [86].

An alternative way to construct a consumption model is to use a physical model for each one of the components on the power chain. Knowing the vehicle speed and acceleration and road topology, one can integrate the dynamic equations and obtain the instant power requirement. Together with physical models, describing for instance tires, transmission, engine mechanics, engine efficiency and fuel chemical properties (for details, refer to [43, 92]), one is able to compute the energy necessary to meet the power requirements. The physical model linking power demand to energy consumption can be obtained, for instance, using empirical maps of engine and/or motor efficiency constructed after a carefully designed set of measurements.

In both kinds of constructions (statistical analysis or physical modeling) the main input variables are the *vehicle speed* and the *road topology*. The vehicle instant acceleration can be estimated by differentiating the speed data. For practical reasons, as mentioned, we do not consider the influence of the road topology in this study, i.e., we neglect the effect of slopes on the energy consumption in our simulations. (We stress that this is not a restrictive simplification as road topology information can be obtained in a deterministic fashion. However, it is clear that any credible consumption estimator must take the topology into account). As a consequence, we treat the speed of the



vehicle as sufficient data to determine the power requirement of the vehicle and thus the energy consumption.

In addition to a consumption model, a power demand profile is needed to assess the energy consumption of the vehicle. To estimate the energy consumption one may therefore use a forecast of the *speed profile*. Once in possession of a speed profile, together with a consumption model, the energy needed for a particular trip can be estimated. This work uses reverse quasi-static models of the components of the vehicle power chain to construct a consumption model (cf. Chapter 1 and [44]). The objective is to capture the essential information of realistic driving patterns to build a model that forecasts representative speed profiles.

As depicted in [36], the external factors influencing driving patterns can be separated in travel behavior, driver factors, vehicle factors, weather factors, traffic conditions and road environment (cf. map in Figure 7.1).

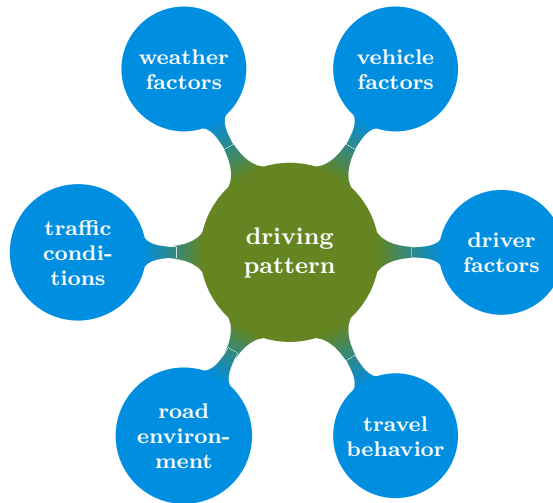


Figure 7.1 – *Diagram of factors affecting driving patterns. Some of these factors can be used to parametrize the energy consumption forecast model. In this study, the energy consumption forecast models are parametrized with respect to the speed recommended by the vehicle navigation system.*

It is pertinent to consider that information is available about some of these factors as to parametrize the forecast model accordingly. More precisely, several forecast models can be constructed, each one corresponding to a set of relevant factors. For example, one can build a forecast model using data sampled from urban driving and another model using data obtained in highways. When an autonomy estimation is required for a known type of road the appropriate forecast model is selected.

Parameterizing several forecast models has the advantage of compartmentalizing factors that play a strong role in the variability of driving patterns. (One can expected a speed profile in a urban environment to somewhat differs from one in a highway). The drawback is that there is less historical data available to the construction of forecast model. In this sense, this work consider vehicle specific power management systems, which fix the vehicle parameters such as mass, aerodynamics characteristics and maximum engine power.

This study also takes into account a navigation system that feeds information about the route environment to the power management system. As explained, the route is divided in links by the navigation system and each link contains its set of information. In particular, to each link is attributed a *suggested cruise speed* corresponding to the speed limit of the street which the link represents minus 10km/h. In this study, we parameterize the forecast models to each of the possible suggested speeds, namely, 20km/h, 40km/h, 60km/h, 80km/h, 100km/h and 120km/h. From a practical point of view, the suggested speed contains relevant information about the link's

surrounding environment and speed limit, as well as street type. Therefore, each forecast model benefits from the compartmentalization of these factors, which are relevant to the variability of the driving pattern [35].

The historical data consists of vehicle speed measurements on a same route. After being separated according to the suggested speed of the link on which the measurements were taken, they are modeled as a stationary time series. To each of the resulting time series, we identify an *autoregressive* (AR) model [23]. To do so, we perform an *a priori* set of analysis on the structure of such data to justify the utilization of AR models of relevant order. After the models are identified, we study their performance with respect to the energy consumption of the vehicle, using the consumption model.

Autoregressive (AR) models are models that seek to describe stationary time series as a linear combination of past realizations. Once a model is identified, one can use it, in principle, to replicate the behavior of the time series within a certain degree of accuracy. This replicated behavior generates data points that can be seen as reproducing the stochastic phenomenon underlying the data presented in the time series.

The choice of AR models for the forecast of speed profiles is justified by their simplicity of implementation and reasonable accuracy, described in detail next. We mention here that there are more sophisticated models, for stationary and non-stationary time series, as for instance ARMAX (ARMA with exogenous inputs), ARIMA (autoregressive integrated moving average), and GARCH (generalized autoregressive conditional heteroskedasticity) models. We avoid many technicals and conceptual details in their description, since the forecast of speed profiles plays only a supportive role in the numerical simulations of power management strategies, which are at the core of this study. We refer to [20, 23] for a detailed description.

Additionally, the term “forecast”, whenever mentioned in this section, does not bear the same meaning understood in the statistics literature. In precise terms, forecasting means using a set of data of a process to best estimate future realizations of the data.

In this work, forecast is used in a broader sense, meaning that one can obtain some information about future events (e.g. the vehicle driven distance) using present information (the historical speed data).

## 7.2 Data Set Analysis and Model Selection

This paragraph describes the data set used for the identification of relevant AR models next.

The historical data consists of instant speed measurements of an electric vehicle. The data is obtained in two different conditions, representing two kind of driving profiles. The first sample set  $S_1$  comprises data obtained from measurements of a *regular* driving profile. Set  $S_1$  represents a home  $\rightleftharpoons$  workplace driving profile of 13.6km being composed of urban/highway/urban segments. The second set used in this study,  $S_2$  represents a relatively less regular driving style aimed to reproduce heavier traffic conditions and/or a more sportive driving style. It is referred to as a *sportive* driving profile. The route used for sampling has a structure of urban/highway/mountain/highway, with a total length of 19.6km. Figure 7.2 shows two examples of speed profile of each sample set. Notice the differences between the more regular profile from set  $S_1$  and the relatively less regular one from set  $S_2$ . In the case of  $S_1$ , the total number of speed profiles available is  $N_1 = 17$  and if  $S_2$ , the number of speed profiles is  $N_2 = 53$ .

As mentioned, each route has an associated recommended speed given by a navigation speed, called *NAV speed*. This particular information from the navigation system is judged to be relevant

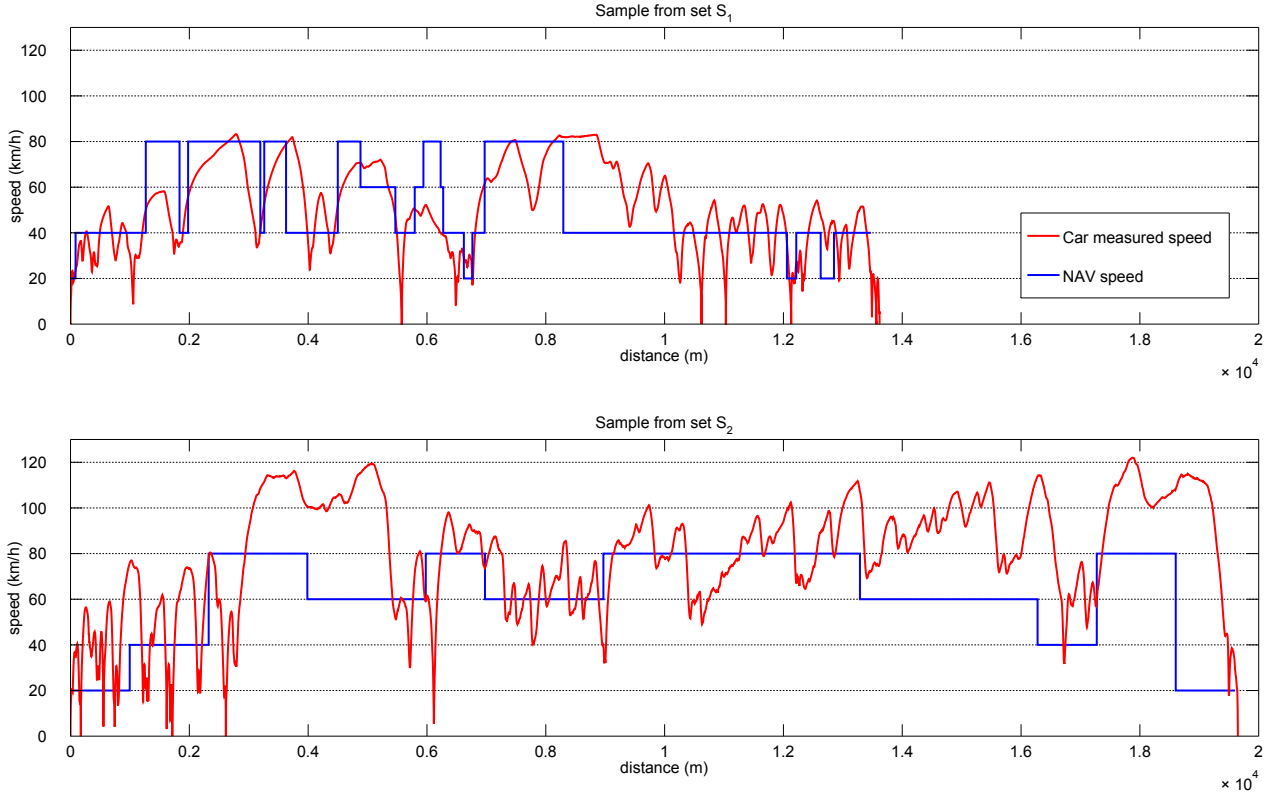


Figure 7.2 – Examples of speed profile data used in the identification of a stochastic speed model. The measurements are shown in red and the associated NAV speed is shown in blue. Above, example of more regular driving profile in a urban/highway/urban route from set  $S_1$ . Below, relatively less regular driving profile in a urban/highway/mountain/highway cycle from set  $S_2$ .

in isolating specific conditions that affect the driving pattern. For that reason, each set data point in  $S_j$ ,  $j = 1, 2$ , is classified according to a particular NAV speed. Proceeding as such, we group data points in subsets of  $S_j$  that share further common features. More precisely, the data from each set  $S_j$  is grouped into one of subsets  $\{\text{NAV20}, \text{NAV40}, \text{NAV60}, \text{NAV80}\}$ , corresponding to the NAV speeds 20km/h, 40km/h, 60km/h, 80km/h. Then, each subset is used to identify an AR model.

We seek to identify an ARMA( $p, q$ ) model of the vehicle speed at each step  $t$ , of the form

$$y_t = \sum_{j=1}^p a_j y_{t-j} + \sum_{j=1}^q b_j e_{t-j+1}, \quad (7.1)$$

where  $e_t$  is a zero-mean white noise of variance  $\sigma_e^2$ . The data in sets  $S_j$ ,  $j = 1, 2$  are connected to form a *time series*, which is denoted by  $\{Y_j\}$ . The goal is to create a stochastic process  $\{y\}$  using (7.1) that has similar statistical properties to  $\{Y_j\}$ .

A first step in the construction of stochastic models is the *sampling time selection*. This is needed in order to select the period of the time delay  $\Delta t = t_j - t_{j+1}$ . The sample time is chosen to be  $\Delta t = 1$ s. This choice is made to enable the stochastic model to capture enough information about the relevant frequencies of the speed dynamics that contributes most to the energy consumption. The greatest contribution for energy consumption in electric vehicles are the breaking phases of the speed profile. A breaking phase denotes the period of time where the driver actually steps on the brake pedal. This causes a dissipation of an important fraction of energy that could otherwise be used to recharge the battery via regenerative breaking. As typical breaking phases take place over a period of  $\approx 5$ s, a sampling time of 1s is judged enough to model this feature.

In order to gain insight about the possible order  $p$  concerning the autoregressive share of the ARMA model, we study the *partial autocorrelation function* of the series  $\{Y\}$ . More precisely (see

[21] for details), denote by  $\gamma_h$  the  $h$ -auto-covariance of  $\{y\}$  (assumed to have zero mean),

$$\gamma_h = \mathbb{E}[y_t y_{t+h}] = \mathbb{E}[y_t y_{t-h}],$$

and by  $\rho_h$  its  $h$ -autocorrelation function,

$$\rho_h = \frac{\gamma_h}{\gamma_0}.$$

The  $h$ -autocorrelation function, valued between 0 and 1, gives the relative degree of dependence of the term  $y_t$  with respect to the past term  $y_{t-h}$ . Figure 7.3 illustrates the relation between the autocorrelation function of lags  $h$  and the autocorrelation matrix (the illustration is inspired from [21, Figure 2.5]).

$$R = \begin{pmatrix} 1 & \rho_1 & \rho_2 & \cdots & \rho_{h-1} \\ \rho_1 & 1 & \rho_1 & \cdots & \rho_{h-2} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \rho_{h-1} & \rho_{h-2} & \rho_{h-3} & \cdots & 1 \end{pmatrix}. \quad (7.2)$$

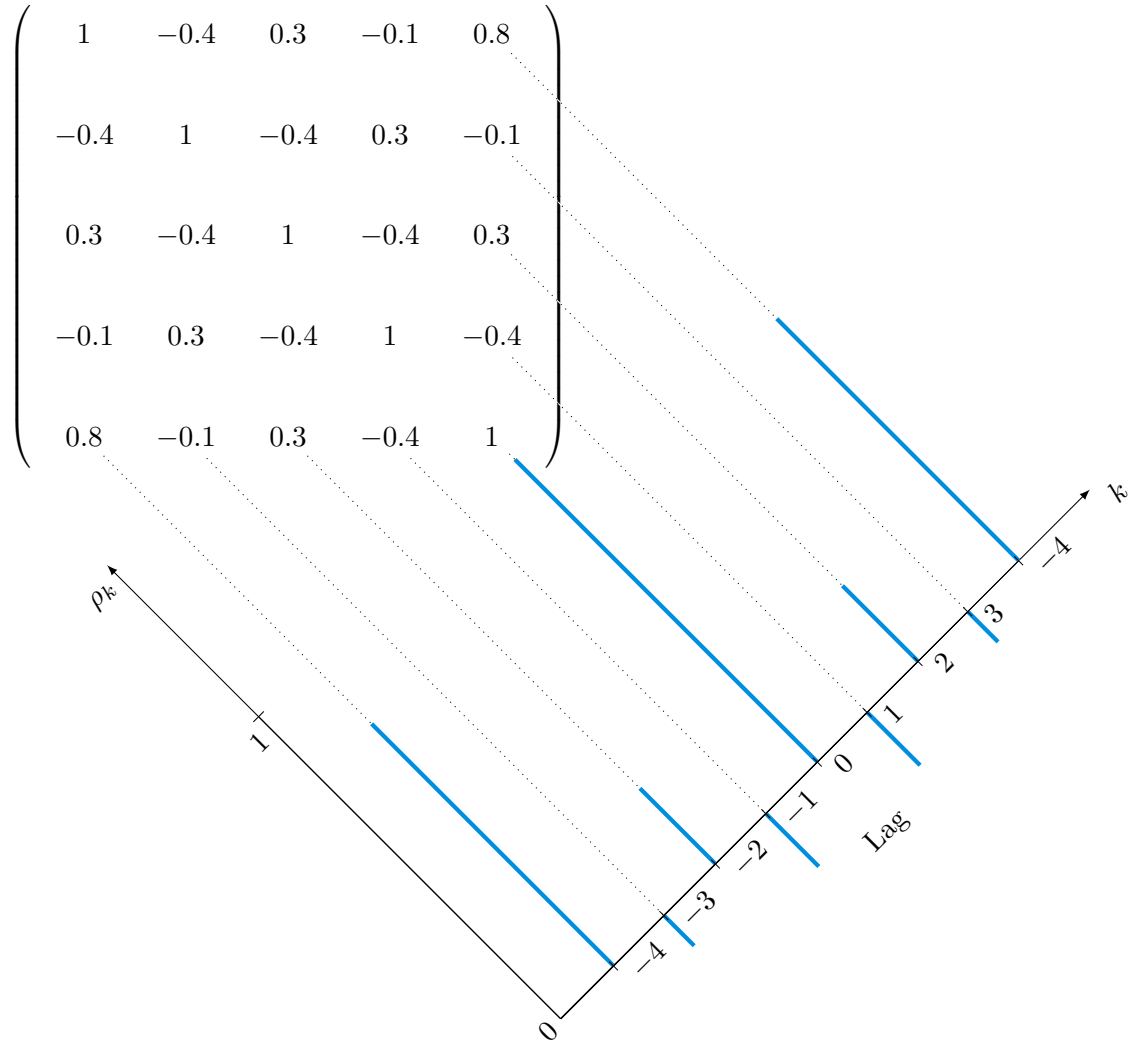


Figure 7.3 – Sketch illustrating the relation between the autocorrelation matrix and the autocorrelation function.

The AR coefficients  $\mathbf{a} := (a_1, \dots, a_h)^\top$  of an autoregressive model of order  $p > 1$ ,  $q = 1$  [23] satisfy a system of equations involving the partial autocorrelation functions  $\boldsymbol{\rho} := (\rho_1, \dots, \rho_h)^\top$  and the autocorrelation matrix  $R$ :

$$R\mathbf{a} = \boldsymbol{\rho}. \quad (7.3)$$

The system of equations (7.3) is known as the *Yule-Walker equations*. Notice that the autocorrelation  $\rho_h$  between the term  $y_t$  and term  $y_{t-h}$  depends on other autocorrelations  $\rho_1, \dots, \rho_{h-1}$ .

The *h-partial autocorrelation function* (PACF) is given by the quantity  $a_h$ . This function gives a degree of the dependence between terms  $y_t$  and  $y_{t-h}$  adjusted for the contribution of intermediary terms  $y_{t-1}, \dots, y_{t-h+1}$ . The PACF vanishes for lags  $h > p$ . Therefore, analyzing the sample PACF of set  $\mathcal{Y}^\#$ , we can infer about the order of the linear autoregressive model that is best suited for representing  $\{Y\}$ .

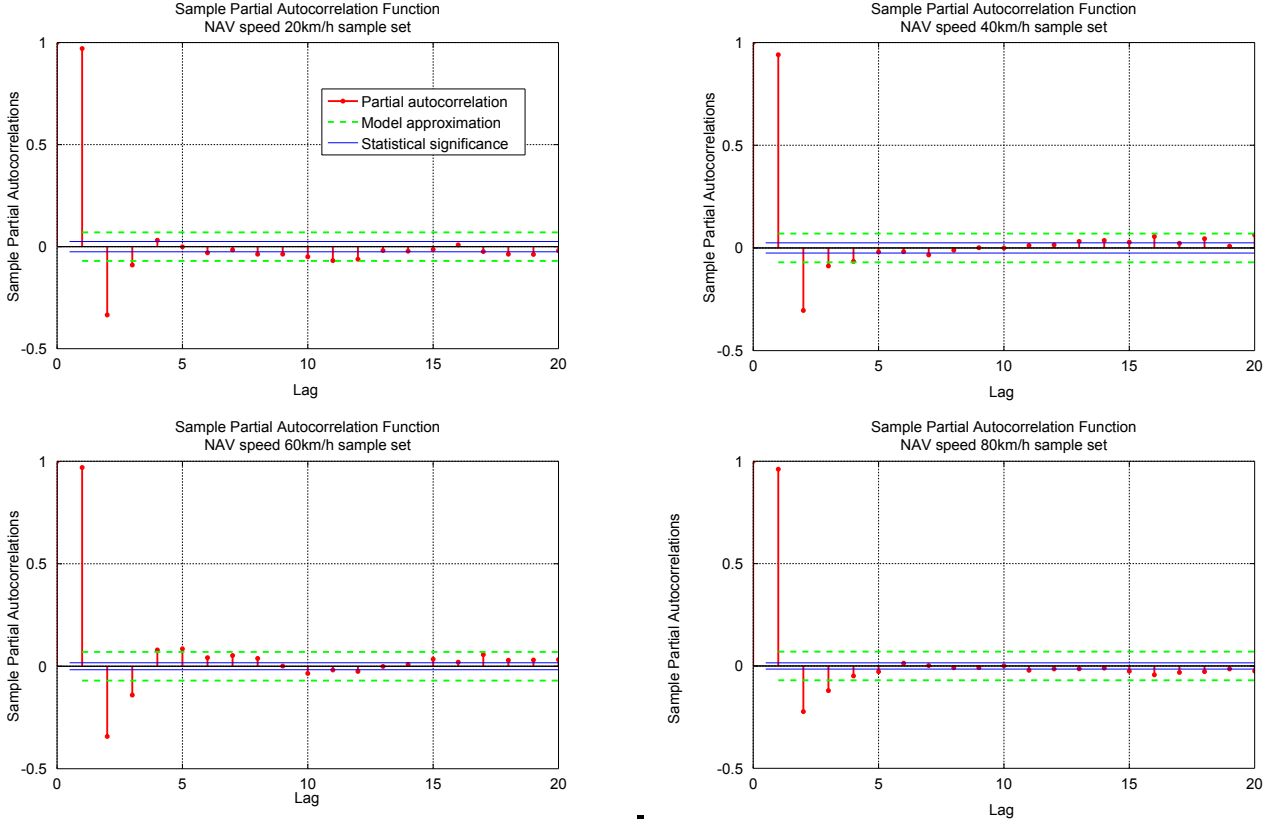


Figure 7.4 – Sample partial autocorrelation functions for the sets of samples obtained from speed measurements. The sampling time used is  $\Delta t = 2s$ . The figure contains the PACF for four sample sets divided according to the corresponding NAV speed present at the moment of the sample. The full blue lines indicate the statistical significance of values of autocorrelations. Values that are between the blue lines are statistically zero. The dashed green lines represent the approximation used in this study. Therefore, correlations of less than 7% are ignored.

Figure 7.4 groups the sample autocorrelation functions of four different sample sets, divided according to the NAV speed, as explained. Each sample set is used to identify one distinct ARMA model. The values of correlation that fall between the two blue lines correspond statistically to zero and are thus ignored.

### 7.3 Consumption Forecast Simulations

Now we turn our attention to the performance of the ARMA models describing the speed evolution as a mean of estimating the energy consumption. As explained, this work uses models describing the physical behavior of the components on the power chain. Using those models, one can input a speed profile  $\Sigma$  and obtain the energy consumption  $E$  associated with the speed profile. This procedure is sketched in Figure 7.5.

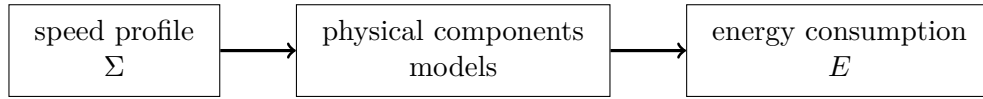


Figure 7.5 – Sketch of the relation between the speed profile generated using an ARMA model, the physical models of the vehicle components and the associated consumption.

To verify the pertinence of the obtained consumption levels using AR generated speed profiles measured speeds profiles obtained in real driving conditions are used.

Associated to each speed profile  $\Sigma_i^j$  from set  $S_j$  there is an energy consumption which we denote by  $E_i^j$ .

The performance of the set of AR models associated with the sample set  $S_i$  is given by the *average relative absolute error with respect to the measured consumption*. More precisely we simulate  $N$  AR speed profiles and denote its associated energy consumption by  $E_k^{\text{ARMA}}$ ,  $k = 1, \dots, N$ . Given a measured driving cycle  $j$  from set  $S_i$ , the associated energy consumption is denoted by  $E_j^{i1}$ .

The average relative absolute error between the measured consumption level of cycle  $j$  from set  $S_i$  and the ARMA consumption levels is then

$$\Delta_j^i = \frac{1}{N} \sum_{k=1}^N \frac{|E_j^i - E_k^{\text{ARMA}}|}{E_j^i}. \quad (7.4)$$

For each set  $S_i$ ,  $i = 1, 2$ , the performance coefficient (or performance factor)  $\eta_i$  of the ARMA model is defined as the average of error defined in (7.4) over all cycles in  $S_i$ :

$$\eta_i = \frac{1}{N_i} \sum_{j=1}^{N_i} \Delta_j^i. \quad (7.5)$$

The average dispersion coefficient is defined as the average over all cycles of the sample standard deviation of the relative absolute error, namely,

$$\sigma_i = \frac{1}{N_i} \sum_{j=1}^{N_i} \sqrt{\frac{1}{N-1} \sum_{k=1}^N \left( \frac{|E_j^i - E_k^{\text{ARMA}}|}{E_j^i} - \Delta_j^i \right)^2} \quad (7.6)$$

Figure 7.6 shows the performance factor for different ARMA models compared against the two measurement sets  $S_1, S_2$ .

From Figure 7.6 one can observe that there is virtually no impact of the order of the model on the relative forecast error for AR orders higher than 3. The effective utilization of information from the navigation system enables the ARMA models to reduce the relative forecast error by roughly 2%. As a general conclusion, one can infer that the ARMA models capture well the features of the driving pattern and can generate energy consumption forecasts within an average 20% of deviation from the real consumption. One can also remark that the model with AR order of 1 presents the best forecast performance of the regular driving profile.

To assess the sensibility of the performance coefficient with respect to the length of the route in which one wishes to estimate the energy consumption, we compare the performance coefficient obtained for cycle of different lengths. In other words, the energy consumption is computed, for a given speed profile, up to a predefined distance. When the vehicle dynamic equations are integrated, the simulation is stopped when the vehicle has reached this predefined distance. Figure 7.7 shows a histogram of the performance coefficient for several values of driven distances.

<sup>1</sup>This consumption level is referred to as "measured" consumption, but we stress that  $E_j^i$  is obtained by the procedure sketched in Figure 7.5 and not actually measured.

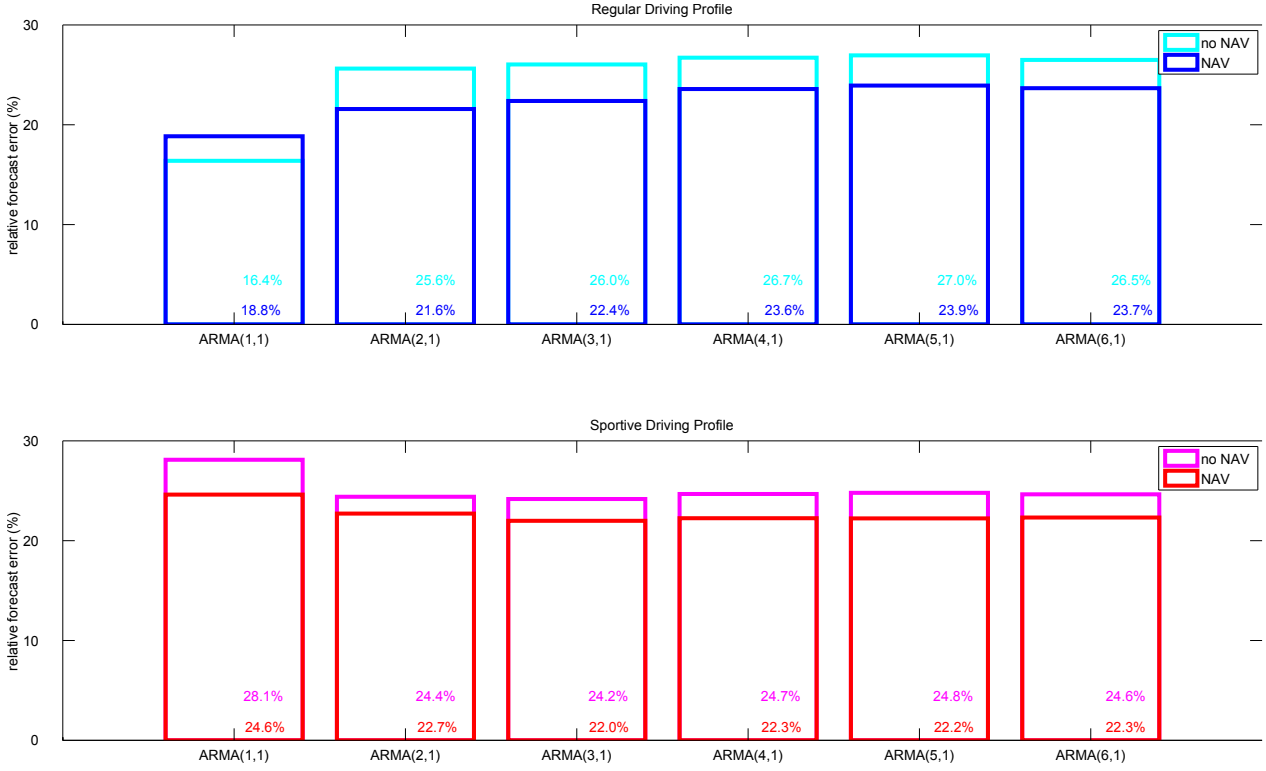


Figure 7.6 – Performance coefficient  $\eta_1, \eta_2$  obtained using 1000 simulations with models to order 6. The figure compares models constructed using exogenous information from the navigation system (NAV, blue, red) against models identified using data irrespectively on NAV information (no NAV, cyan, pink).

One notices that there is little impact on the forecast error due the driven distance used in the simulations, except for the sportive profile using 5km.

The identification of ARMA models to describe the behavior of the vehicle speed modeled as a random variable represents a first approximation towards the formulation of the optimal control problem that is the aim of this work. Indeed, to formulate the stochastic optimal control problem one requires what we shall refer to as an *optimization model*. This optimization model corresponds, as all models do, to a simplification of the "real" physical phenomenon. In this sense, as the evolution of the "real" vehicle speed is unlikely to be described with certainty, we describe its evolution by a random process where the random variable verifies one of the ARMA models analyzed studied above.

## 7.4 Markov Chain Transition Probability Estimation

In this subsection we construct the probability law of the random variable modeling the vehicle's future speeds. We use a kernel density estimation method on a sample space obtained through simulations using the ARMA models.

The optimization model used to formulate the stochastic optimal control problem for the synthesis of optimal power management strategies considers the future vehicle speed as a random variable. As explained in Chapter 1, the vehicle speed at a future node on the vehicle path needs to be considered. More precisely, we assume that we have complete information about the *current vehicle speed*, denoted  $y$ , i.e., we assume we can measure with enough precision the vehicle speed. Then, the vehicle speed at the next node is modeled as a random variable  $\xi$ .

To properly define this random variable a *probability space* is constructed (for a fuller study on

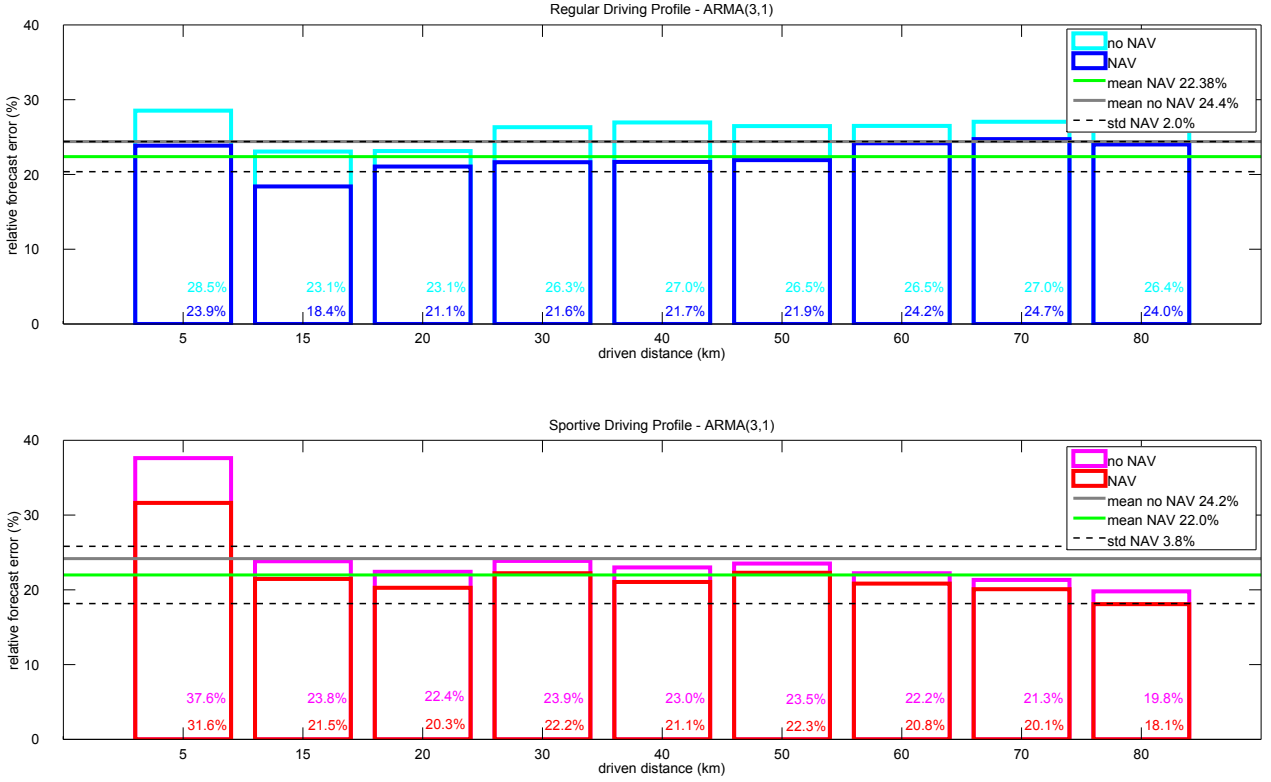


Figure 7.7 – Performance factors  $\eta_1, \eta_2$  and associated standard deviation obtained using 1000 ARMA(3,1) simulations for cycles of different lengths. The cycles are from the different data sets  $S_1, S_2$ . For simulated lengths greater than the cycle, cycles are concatenated.

random variables and practical issues see [13]). From a theoretical point of view, the construction of such probability space poses no particular problem. From a practical point of view however, one has to construct a probability law  $\mathbb{P}$  over the  $\sigma$ -algebra of the set of possible speeds.

Denote by  $y, \bar{y}$  the lower and upper bounds on the vehicle speed, respectively. Define  $\Xi := [y, \bar{y}]$ , the interval of possible admissible speed values,  $\xi \in \Xi$ . Next, define a  $\sigma$ -algebra over  $\Xi$  denoted by  $\mathcal{F}$ , a probability  $\mathbb{P}$  and a probability space  $(\Xi, \mathcal{F}, \mathbb{P})$ . The probability law  $\mathbb{P}$  quantifies the possible occurrences of the random variable  $\xi$ .

Now, consider a discrete process  $\{\xi\} := (\xi_1, \dots, \xi_T)$  of random variables  $\xi$  and denote by  $\mathcal{F}_t$  the filtration of  $\{\xi\}$  at time  $t$ . Given a process  $\{\xi\}$  we are interested in describing the probabilities of  $\xi_t$  conditional to  $\mathcal{F}_{t-1}$ . In particular, this information includes the current and past vehicle speed. This speed process is modeled as a Markov chain. In view of a practical consideration, the stochastic optimal control problem is solved via a stochastic dynamic programming algorithm. Therefore, due the curse of dimensionality, it is well advised to avoid increasing to much (beyond 6) the state dimension. A Markov chain contributes with one state variable<sup>1</sup>.

To simplify notation, denote by  $y$  the last known value of process  $\{\xi\}$  and by  $\xi$  the next value, i.e., given  $\mathcal{F}_t$ ,  $y := \xi_t$  and  $\xi := \xi_{t+1}$ . To describe the Markov chain process, the *transition probabilities*  $p(y, \xi) := \mathbb{P}(\xi | y)$  need to be identified.

Two approaches may be considered in identifying probability densities, namely, using parametric or non-parametric models. Parametric modeling consists of choosing an *a priori* parametric density function (e.g. Gaussian, exponential, Cauchy) and determining its parameters, for instance, using a best least-square fit of the distribution and the sampled data. This approach can be justified

<sup>1</sup>Briefly and roughly, the *curse of dimensionality* labels the fact that a dynamic programming algorithm requirements grows exponentially with the number of past speeds  $\xi_t, \xi_{t-1}, \dots, \xi_{t-k}$  considered in the model of future speed values  $\xi_{t+1}$ .



by knowing that the sample data (population) are drawn from a known parametric distribution. On the contrary, non-parametric methods do not impose any particular structure of the density function beforehand and thus can be seen as more robust (cf. [46]). A histogram is an example of a non-parametric estimation of a probability density.

In this work we use a *kernel density estimation* (KDE) method for constructing a smooth density probability function (see [83, 82] for a more detailed description). The KDE is a non-parametric method.

Given a sample set, the main idea is to place, centered at each observation, a given probability density. More precisely, consider a data set  $\mathcal{Y}^\#$  containing  $M$  observations of the form  $\mathbf{y}_j := (y_j, \xi_j) \in \Xi \times \Xi$ . Consider a two-dimensional density kernel  $K : \Xi \times \Xi \rightarrow [0, 1]$  and a smoothing parameter  $h > 0$ . Given a particular observation  $\mathbf{y}_j = (y_j, \xi_j)$ , one approximates locally the density of  $\xi$  conditional to  $y$  by adding the density kernel centered on the observation  $\mathbf{y}_j$ :

$$\frac{1}{h^2} K\left(\frac{\mathbf{y}_j}{h}\right).$$

Then, the estimator of the density of the conditional probability function of  $\xi$  given  $y$ , denoted by  $\hat{p}(\mathbf{y}) := \hat{p}(y, \xi)$ , is the normalized contribution of all  $M$  points of the data set  $\mathcal{Y}^\#$

$$\hat{p}(\mathbf{y}) = \frac{1}{Mh^2} \sum_{j=1}^M K\left(\frac{\mathbf{y} - \mathbf{y}_j}{h}\right). \quad (7.7)$$

For this study, the density kernel is chosen to be a bivariate Gaussian kernel

$$K(x) := \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}}.$$

We use a bivariate kernel density estimator as described in [18] (the Matlab<sup>®</sup> function for univariate and bivariate kernel estimator identification is available on-line).

At this point, we give some technical considerations of our vehicle application. We recall that we seek to use speed profiles in order to identify a density probability function of a future speed  $\xi$  having knowledge of a current speed  $y$ .

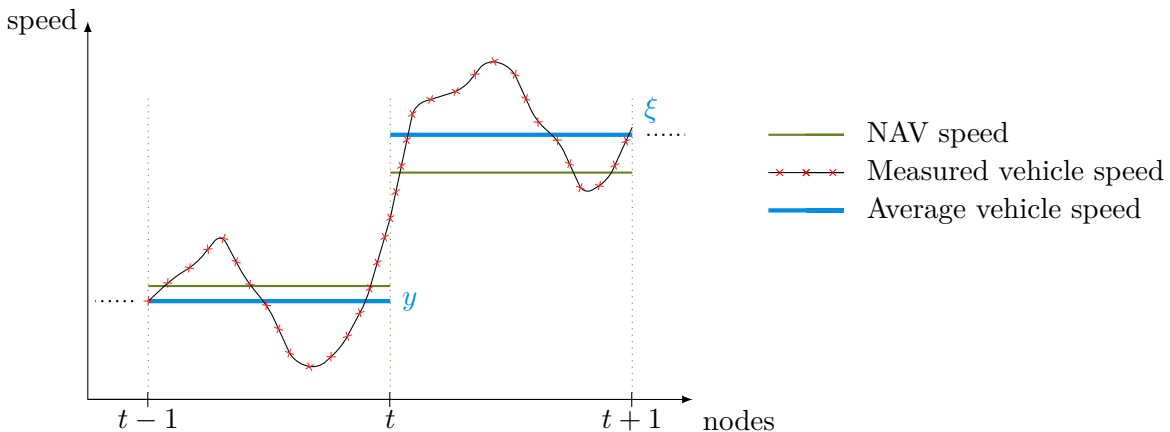


Figure 7.8 – Figure illustrating the information contained in the data set used for probability estimation. The 16 estimators are constructed according to the NAV speed (green) between links. The sample points contain the values of the average (blue) of the measured speeds over a link.

The first point is to give a precise meaning of where in the “future” the random variable  $\xi$  is. The controller acts only at certain points of the vehicle route. The future speed is considered to be

the speed at the next point of the route on which the controller will act. In terms of the vocabulary introduced in Chapter 1, considering a route segmented composed of *links* and *nodes*, the current speed  $y$  is associated with the vehicle speed at the current node whereas the future speed  $\xi$  is associated with the vehicle speed at the *next* node.

The second observation is that, like the ARMA model identification procedure, it is interesting to condition the set  $\mathcal{Y}^\#$  to available information in the navigation system seeking therefore to construct more representative estimators. Since  $\hat{\mathbf{p}}$  is a bivariate estimator, one needs to consider the NAV speed *at both* sample points  $y$  (current NAV speed) and  $\xi$  (NAV speed at next link). In other words, we seek to construct one estimator  $\hat{\mathbf{p}}$  per NAV speed combinations. In our case, this amounts to  $4^2 = 16$  estimators.

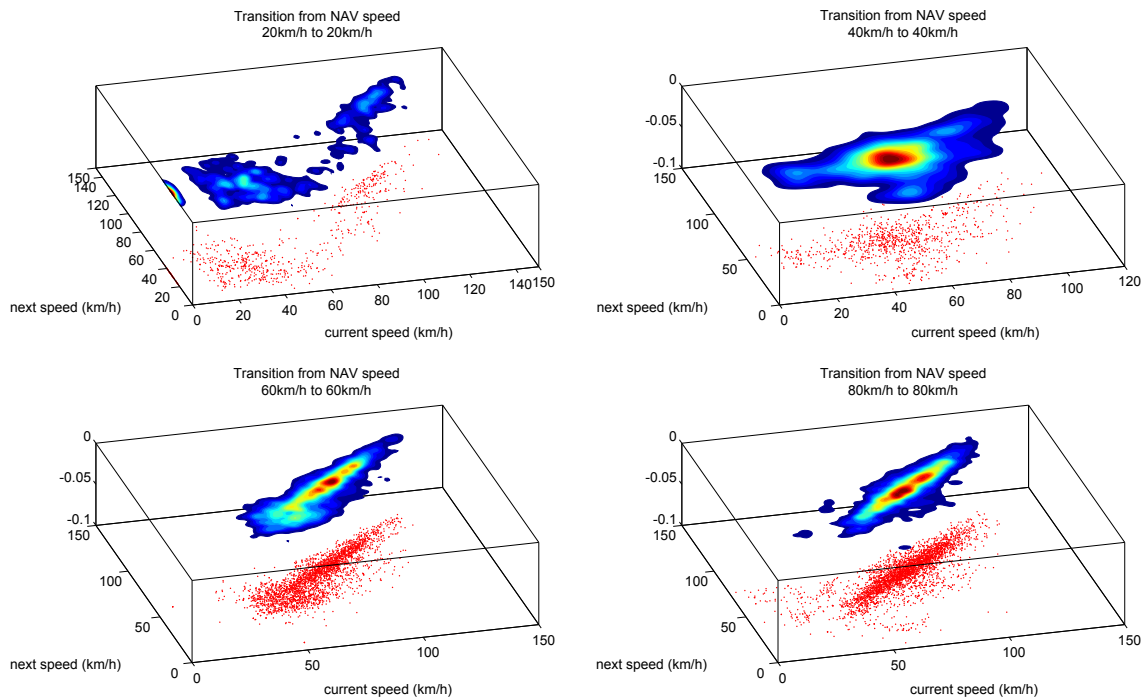
A third remark concerns what information the sample points actually encode. The speed sample  $y$  at node  $t$  is the *average speed* of the instant speed measures between nodes  $t - 1$  and  $t$ . Similarly, the sample speed  $\xi$  at node  $t + 1$  is the average speed between nodes  $t$  and  $t + 1$ . Figure 7.8 illustrates the above statements.

The last remark refers to what speed profiles are employed to create the observations used in the construction of the estimator defined in 7.7. One can use directly the real speed data available from driving measurements. On the other hand, in possession of ARMA models capable of simulating the speed process  $\{\xi\}$ , one can generate as many process as necessary to obtain a relatively greater number of observations in  $\mathcal{Y}^\#$ . The first approach has the advantage of having the sample points draw from the "real" distribution governing the stochastic process  $\{\xi\}$ . Using the measured data raises two points:

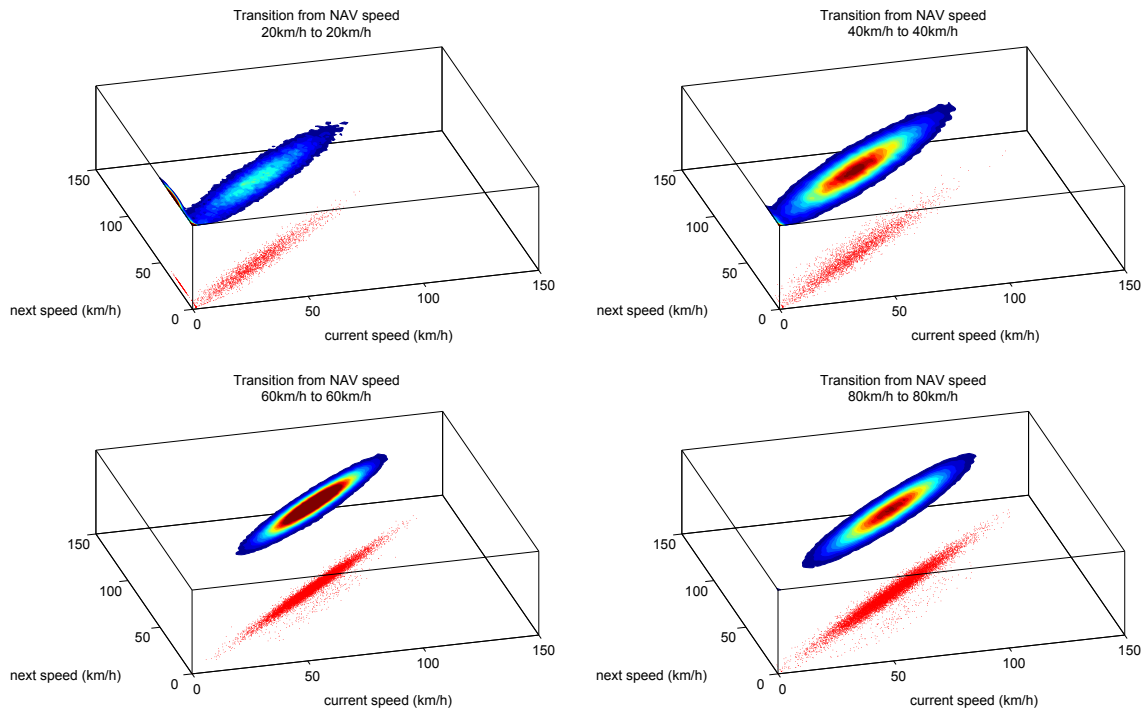
- **Quality of the estimators.** Firstly, one is limited to only a relatively small number of samples. Indeed, as mentioned, this study has 53 driving records available and the observations are distributed among the 16 existent NAV speeds transitions, reducing further the number of sample points available to be used in the identification of each estimator.
- **Relevance of the estimators.** Secondly, the simulations to assess the efficiency of optimal controllers use speed profiles generated with the autoregressive models identified with the measured data. This means that, considering probabilities identified from measured data, the optimal control problem synthesizes controllers that take into account information about the "real" process. However, the performance of the controllers are assessed using simulated processes generated from an autoregressive model, hence, not constituting the same processes that synthesized the controller. In the light of this conceptual incoherence, it is not clear how one could interpret and draw conclusions about the quality of control strategies from the simulation results.

To shed a light in the points explained above, we construct the transition probabilities estimators with both data obtained from autoregressive models simulations and measured data. Proceeding as such, the differences between stochastic control strategies associated to the different estimators can be investigated.

Figure 7.9 shows four of the sixteen estimators  $\hat{\mathbf{p}}$  obtained through the kernel density estimator method. Each set of 16 estimators is identified using two different sample sets. The first contains speed measurements obtained from real driving data. The second contains speed values obtained from speed profiles simulated using an ARMA(2,1) model. Qualitatively, one can observe data the estimators obtained from measured data are relatively more dispersed than from the estimators obtained from simulated data. In this work, we do not perform any statistical analysis on the obtained probability density estimators. Their pertinence is analyzed in the light of the performance of the associated power management strategies described in Chapter 2.



(a) Measured driving cycles.



(b) Simulated driving cycles

Figure 7.9 – Four examples of estimators  $\hat{p}$  of the density probability obtained using the measured data set from 53 driving profiles (a) and from sample set constructed using ARMA simulated speed profiles (b). From the upper-leftmost proceeding clockwise, they represent the estimators for NAV speed transition of 20km/h  $\rightarrow$  20km/h, 40km/h  $\rightarrow$  40km/h, 60km/h  $\rightarrow$  60km/h and 80km/h  $\rightarrow$  80km/h respectively. The red dots are the sampled points of set  $\mathcal{Y}^\#$  whereas the filled surface indicates the probability density.

---

# CONCLUSION AND PERSPECTIVES

---

*“ Wherefore I perceive that there is nothing better, than that a man should rejoice in his own works; for that is his portion: for who shall bring him to see what shall be after him?”*

**The Holy Bible, KJV, Ecclesiastes 3:22.**

The techniques detailed in this document seek to enhance the efficiency of power management of hybrid vehicles. By exploiting the greater degree of freedom created by the introduction of an extra power source, optimal control techniques offer a large range of tools to do so. This study contributes to the engineering community by investigating some of these available tools and techniques.

They may be categorized according to the level of information available for the synthesis of the controller. In one case, often referred to as *deterministic case*, the controller has access to all information needed for optimality with perfect certainty. In the other case, called *stochastic case*, the controller has perfect information only about the past of the system. Information about future events is available, but not with certainty.

In the deterministic configuration, the optimization of the financial cost is fairly well documented in the literature. We suggest an interesting approach to two wholly new applications: the optimization of the vehicle autonomy and the optimization of the flexibility over the final energy distribution. In a continuous-time setting, this work contributes by introducing new theoretical results, namely, the proof of a comparison principle for the value function associated to the optimal control of a hybrid dynamical system and the proof of the convergence of a class of finite-difference numerical schemes. This results enables the numerical computation of the value function, effectively allowing the range/flexibility optimization.

In addition, the approach is considered in a discrete-time setting, with similar goals. The reachability problem is originally stated as an obstacle problem, instead of a state-constrained optimal control problem. The numerical algorithm arising from the discrete-time setting analysis

offers the advantage of being of more intuitive implementation and calibration, being thus, closer to engineering needs in developing real-life controllers.

The study of the stochastic setting provides interesting insight on how real-life control strategies can be made more economic and cleaner. Firstly, by introducing an adequate stochastic model for driving profiles using information from the navigation system, we are able to construct an adequate simulation platform. This platform allows for the simulation of stochastic strategies and their subsequent interpretation. Two numerical approaches to compute stochastic control policies are studied. The first is a stochastic dynamic programming algorithm and the second a stochastic dual dynamic programming. In the first case, the performance of the resulting stochastic controllers is very satisfactory. In particular, the stochastic strategies present are somewhat robust, assuring the success of the mission in spite of imperfect information. At the same time, these strategies outperform two heuristically designed strategies. These performance gains present a potential for substantial cost and emissions reductions over a mid- or long-term utilization of a hybrid vehicle.

With the objective of reducing the computational needs required to compute the value function, a potential application of a stochastic dual algorithm programming (SDDP) is studied. In particular, this work addresses the main bottleneck to the application of the SDDP to the optimal control problem: the model convexity properties. We propose a convex non-hybrid approximation of the original hybrid dynamical model. This new version of the model is crafted to fit as best as possible the main share of scenarios where a stochastic strategy is expected to be applied.

The optimal stochastic power management offers a new interesting application to another optimal control technique, namely, the study of chance constrained optimization problems. Here, the interest is in the introduction of the right notion of chance constrained power management. We construct an adequate problem formulation and investigate its information structure to derive a dynamic programming principle. The dynamic programming principle can, in turn, stem an algorithm to compute optimal policies that are more flexible than their almost-surely constrained counterparts.

The results and ideas presented in this document offer a rich spectrum of new promising applications and further research topics. We number a few, more readily exploitable, interesting examples:

- **Stochastic reachability.** The natural extension of the reachability study developed here is the introduction of a stochastic setting. In this case, the reachable set may be defined as the probability of reaching some energy state. In consequence, one can compute a probability distribution of maximum vehicle range. This information can be used to propose to the driver a driver profile that enhances his/her chances of reaching the furthest possible point. Alternatively, if the driver chooses the final state, he/she can do it within a certain confidence level.
- **Advanced driving profile stochastic models.** An effort can be made towards the implementation of more evolved stochastic models of the driving profile. More sophisticated stochastic modeling may capture in more detail the features of the real measured driving profiles. If these models can well reproduce (stochastically) the real process underlying the measurements, they can be used in the approach proposed in this study: generate artificial

data to feed optimization models, simulation ends and other applications. This procedure may benefit from a larger database of driving measurements.

- **More complex vehicle models.** The introduction of more complex models for modeling the vehicle seems a very relevant consideration. For instance, more refined battery models may offer the possibility of synthesizing controllers to take into account the temperature or the state of health of the battery. One can also think of optimizing the duration of the battery life, since the high-voltage battery is by far the most important (and expensive!) component in the power train.
- **Greater driver interaction.** With the capabilities of incorporating additional information in the synthesis of optimal controllers, the driver may play a pivotal role in further enhancing the performance of stochastic strategies. The driver can participate in this process by providing him/herself additional information to the controller (e.g. driving style, particular conditions not present in the navigation system, customized preferences). In the other direction, the machine may communicate relevant information to the driver, expecting to modify his/her behavior towards more efficient driving conditions. On-line information in this sense can be in the form of probabilities of failure, suggesting more efficient routes, notifying changes in driver driving style. Off-line, the driver may participate in social communities comparing emission or energy efficiency data with others or receive customized feedback on how improve his/her driving style.

We hope that all the modest contributions present in this document eventually help the advance of electrified power train technology. If not, may they awake the interest for the subject in other researchers so to push a bit further the boundaries of engineering and mathematics.

Probably, in the future, the good things in life will continue to be somewhere else. Hopefully, we will get there in a cleaner car.



---

# BIBLIOGRAPHY

---

- [1] United States Energy Information Administration. Annual energy outlook 2012. <http://www.eia.gov/forecasts/aeo/>, 2012.
- [2] European Environment Agency. Co2 emissions performance of car manufacturers in 2010. Technical report, European Environment Agency.
- [3] K. Ahn, M. Van Aerde, et al. Estimating vehicle fuel consumption and emissions based on instantaneous speed and acceleration levels. *Journal of Transportation Engineering*, 128:182, 2002.
- [4] L. Andrieu, G. Cohen, and F. Vázquez-Abad. Stochastic programming with probability. *Arxiv preprint arXiv:0708.0281*, 2007.
- [5] A. Arutyunov, V. Dykhta, and F. Lobo Pereira. Necessary conditions for impulsive nonlinear optimal control problems without a priori normality assumptions. *Journal of Optimization Theory and Applications*, 124:55–77(23), January 2005.
- [6] V. Bally and G. Pagès. Error analysis of the optimal quantization algorithm for obstacle problems. *Stochastic Processes and their Applications*, 106(1):1–40, 2003.
- [7] V. Bally and G. Pagès. A quantization algorithm for solving multidimensional discrete-time optimal stopping problems. *Bernoulli*, 9(6):1003–1049, 2003.
- [8] M. Bardi and I. Capuzzo-Dolcetta. *Optimal Control and Viscosity Solutions of Hamilton-Jacobi-Bellman Equations*. 1997.
- [9] G. Barles, S. Dharmatti, and M. Ramaswamy. Unbounded viscosity solutions of hybrid control systems. *ESAIM: Control, Optimization and Calculus of Variations*, 16:176–193, 2010.
- [10] G. Barles and P. E. Souganidis. Convergence of approximation schemes for fully nonlinear second order equations. *Asymptotic Anal.*, 4(3):271–283, 1991.
- [11] R. Bellman. On the theory of dynamic programming. *Proceedings of the National Academy of Sciences of the United States of America*, 38(8):716, 1952.
- [12] R.E. Bellman and S.E. Dreyfus. *Applied dynamic programming*, volume 7962. Princeton University Press Princeton, NJ, 1964.



- [13] M. Benaïm and N. El Karoui. *Promenade aléatoire: chaînes de Markov et simulations; martingales et stratégies*. Ecole Polytechnique, 2004.
- [14] A. Bensoussan and J. L. Menaldi. Hybrid control and dynamic programming. *Dynamics of Continuous Discrete and Impulsive Systems*, (3):395–442, 1997.
- [15] D.P. Bertsekas. Dynamic programming and optimal control. 1995.
- [16] O. Bokanowski, A. Briani, and H. Zidani. Minimum time control problems for non-autonomous differential equations. *System and Control Letters*, 58:742–746, 2009.
- [17] O. Bokanowski, N. Forcadel, and H. Zidani. Reachability and minimal times for state constrained nonlinear problems without any controllability assumption. *SIAM J. Control and Optimization*, 48(7):4292–4316, 2010.
- [18] Z. I. Botev, J. F. Grotowski, and D. P. Kroese. Kernel density estimation via diffusion. *The Annals of Statistics*, 38(5):2916–2957, 2010.
- [19] B. Bouchard, N. Touzi, R. Elie, et al. Stochastic target problems with controlled loss. *SIAM Journal on Control and Optimization*, 48(5):3123–3150, 2009.
- [20] G. E. P. Box, G. M. Jenkins, and G.C. Reinsel. *Time series analysis*. Holden-day San Francisco, 1976.
- [21] G. E. P. Box, G.M. Jenkins, and G. C. Reinsel. *Time Series Analysis, Forecasting and Control, 3rd edition*. Prentice Hall, Englewood Cliffs, NJ, 1994.
- [22] M. S. Branicky, V. S. Borkar, and S. K. Mitter. A unified framework for hybrid control: Model and optimal control theory. *IEEE Transactions On Automatic Control*, 43:31–45, 1998.
- [23] P. J. Brockwell and R. A. Davis. *Introduction to Time Series and Forecasting*. Springer Texts in Statistics. Springer, 2002.
- [24] B. Bruder and H. Pham. Impulse control problem on finite horizon with execution delay. *Stochastic Processes and their Applications*, 119:1436–1469, 2009.
- [25] P. E. Caines, F. H. Clarke, X. Liu, and R.B. Vinter. A maximum principle for hybrid optimal control problems with pathwise state constraints. In *45th IEEE Conference on Decision and Control*, pages 4821 –4825, dec. 2006.
- [26] I. Capuzzo-Dolcetta and L. C. Evans. Optimal switching for ordinary differential equations. *Siam Journal On Control And Optimization*, 22(1):143–161, 1984.
- [27] P. Cardaliaguet, M. Quincampoix, and P. Saint-Pierre. Optimal times for constrained nonlinear control problems without local controllability. *Appl. Math. Opt.*, 36:21–42, 1997.
- [28] P. Carpentier, J.P. Chancelier, G. Cohen, M. De Lara, and P. Girardeau. Dynamic consistency for stochastic optimal control problems. *Annals of Operations Research*, pages 1–17, 2010.
- [29] A. Charnes, W.W. Cooper, and G.H. Symonds. Cost horizons and certainty equivalents: an approach to stochastic programming of heating oil. *Management Science*, 4(3):235–263, 1958.
- [30] M. G. Crandall, H. Ishii, and P-L. Lions. User’s guide to viscosity solutions of second order partial differential equations. *Bull. Amer. Math. Soc. (N.S)*, (27), 1992.
- [31] M.G. Crandall and P.L. Lions. Viscosity solutions of Hamilton-Jacobi equations. *Transactions of the American Mathematical Society*, 227(1):1–42, 1983.
- [32] S. Dharmatti and M. Ramaswamy. Hybrid control systems and viscosity solutions. *SIAM J. Control Optim.*, 44(4):1259–1288, 2005.
- [33] S. Dharmatti and M. Ramaswamy. Zero-sum differential games involving hybrid controls. *Journal Of Optimization Theory And Applications*, 128(1):75–102, Jan 2006.

- [34] J. Dupačová, G. Consigli, and S.W. Wallace. Scenarios for multistage stochastic programs. *Annals of operations research*, 100(1):25–53, 2000.
- [35] E. Ericsson. Driving pattern in urban areas - descriptive analysis and initial prediction model. 2000.
- [36] E. Ericsson. Variability in urban driving patterns. *Transportation Research Part D: Transport and Environment*, 5(5):337–354, 2000.
- [37] Eurostat. Transport energy consumption and emissions. [http://epp.eurostat.ec.europa.eu/statistics\\_explained/index.php/Transport\\_energy\\_consumption\\_and\\_emissions#Further\\_Eurostat\\_information](http://epp.eurostat.ec.europa.eu/statistics_explained/index.php/Transport_energy_consumption_and_emissions#Further_Eurostat_information), 2006.
- [38] Eurostat. Panorama of transport. [http://epp.eurostat.ec.europa.eu/cache/ITY\\_OFFPUB/KS-DA-09-001/EN/KS-DA-09-001-EN.PDF](http://epp.eurostat.ec.europa.eu/cache/ITY_OFFPUB/KS-DA-09-001/EN/KS-DA-09-001-EN.PDF), 2009.
- [39] United Nations Economic Commission for Europe. Application of ECE regulations to hybrid vehicles. *42nd GRPE, Informal document 3, agenda item 8*, 2001.
- [40] M. Garavello and B. Piccoli. Hybrid necessary principle. *SIAM Journal On Control And Optimization*, 43(5):1867–1887, 2005.
- [41] P. Girardeau. *Résolution de grands problèmes en optimisation stochastique dynamique et synthèse de lois de commande*. PhD thesis, Université Paris-Est, 2010.
- [42] F. E. Grubbs. Procedures for detecting outlying observations in samples. *Technometrics*, 11:1–21, 1969.
- [43] L. Guzzella and C. H. Onder. *Introduction to modeling and control of internal combustion engine systems*. Springer Verlag, 2010.
- [44] L. Guzzella and A. Sciarretta. *Vehicule Propulsion Systems, Introduction to Modeling and Optimization. Second edition*. Springer, 2007.
- [45] H. Heitsch and W. Römisch. Scenario tree modeling for multistage stochastic programs. *Mathematical Programming*, 118(2):371–406, 2009.
- [46] J. L. Horowitz. *Semiparametric and nonparametric methods in econometrics*. Springer Verlag, 2009.
- [47] T. Huria, M. Ceraolo, J. Gazzarri, and R. Jackey. High fidelity electrical model with thermal dependence for characterization and simulation of high power lithium battery cells. In *Electric Vehicle Conference (IEVC), 2012 IEEE International*, pages 1–8. IEEE, 2012.
- [48] L. Johannesson, M. Asbogard, and B. Egardt. Assessing the potential of predictive control for hybrid vehicle powertrains using stochastic dynamic programming. *Intelligent Transportation Systems, IEEE Transactions on*, 8(1):71–83, 2007.
- [49] V.H. Johnson, A.A. Pesaran, T. Sack, National Renewable Energy Laboratory (US), and S. America. *Temperature-dependent battery models for high-power lithium-ion batteries*. National Renewable Energy Laboratory, 2001.
- [50] G.V. Kaiper. Us energy flow-1999. Technical report, Lawrence Livermore National Laboratory report no. UCRL-ID-129990-99. Livermore, CA: March, 2001.
- [51] A.J. Kleywegt, A. Shapiro, and T. Homem-de Mello. The sample average approximation method for stochastic discrete optimization. *SIAM Journal on Optimization*, 12(2):479–502, 2002.
- [52] I. Kolmanovsky, I. Siverguina, and B. Lygoe. Optimization of powertrain operating policy for feasibility assessment and calibration: Stochastic dynamic programming approach. In *American Control Conference, 2002. Proceedings of the 2002*, volume 2, pages 1425–1430. IEEE, 2002.

- [53] M.A. Kromer and J.B. Heywood. *Electric powertrains: opportunities and challenges in the US light-duty vehicle fleet*. PhD thesis, Massachusetts Institute of Technology, Engineering Systems Division, Technology and Policy Program, 2007.
- [54] Argonne National Laboratory. The greenhouse gases, regulated emissions, and energy use in transportation model. <http://greet.es.anl.gov/>, 2012.
- [55] O. Ley. Lower-bound gradient estimates for first-order hamilton-jacobi equations and applications to the regularity of propagating fronts. *Adv. Differential Equations*, 6(5):547–576, 2001.
- [56] C.C. Lin, H. Peng, and JW Grizzle. A stochastic control strategy for hybrid electric vehicles. In *American Control Conference, 2004. Proceedings of the 2004*, volume 5, pages 4710–4715. IEEE, 2004.
- [57] J. Lin and D.A. Niemeier. An exploratory analysis comparing a stochastic driving cycle to california’s regulatory cycle. *Atmospheric Environment*, 36(38):5759–5770, 2002.
- [58] J. Lygeros, D. N. Godbole, and S. Sastry. Verified hybrid controllers for automated vehicles. *IEEE Transactions On Automatic Control*, 43:522–539, 1996.
- [59] J. Lygeros, C. Tomlin, and S. Sastry. Controllers for reachability specifications for hybrid systems. *Automatica*, 35:349–370, 1999.
- [60] B.L. Miller and H.M. Wagner. Chance constrained programming with joint constraints. *Operations Research*, pages 930–945, 1965.
- [61] 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. *Control Systems Technology, IEEE Transactions on*, (99):1–11, 2010.
- [62] M. D. Murphy. Fuel consumption estimation. Patent, 06 1999. US 5913917.
- [63] Office of the Law Revision Counsel. 15 USCS § 2502 (5), Title 15. Commerce and Trade; Chapter 52. Electric and Hybrid Vehicle Research, Development, and Demonstration. <http://uscode.house.gov/download/pls/15C52.txt>, 2012.
- [64] S. Osher and J.A. Sethian. Fronts propagating with curvature dependent speed: Algorithms based on Hamilton-Jacobi Formulations. *Journal of Computational Physics*, 79:12–49, 1988.
- [65] S. Osher and C-W. Shu. Higher order essentially non-oscillatory schemes for Hamilton-Jacobi equations. *SIAM Journal on Numerical Analysis*, 28(4):907–922, August 1991.
- [66] P.R. Pacheco. Study of real-time estimation techniques related to the autonomy of an electric vehicle. 2011.
- [67] M.V.F Pereira and L.M.V.G. Pinto. Multi-stage stochastic optimization applied to energy planning. *Mathematical Programming*, 52:359–375, 1991.
- [68] L.S. Pontryagin, V.G. Boltyanskii, R.V. Gamkrelidze, and E.F. Mishchenko. The mathematical theory of optimal processes.(translated from the russian) Interscience Publishers John Wiley & Sons. Inc., New York–London, 1962.
- [69] A. Prékopa. On probabilistic constrained programming. In *Proceedings of the Princeton Symposium on Mathematical Programming*, pages 113–138. Citeseer, 1970.
- [70] A. Prékopa. Contributions to the theory of stochastic programming. *Mathematical Programming*, 4(1):202–221, 1973.
- [71] A. Prékopa. *Stochastic programming*, volume 324. Springer, 1995.
- [72] R.T. Rockafellar and R.J.B. Wets. *Variational analysis*, volume 317. Springer Verlag, 1998.

- [73] G. Rousseau. *Véhicules Hybrides et Commande Optimale*. PhD thesis, École des Mines de Paris, 2008.
- [74] A. Ruszczyński. Risk-averse dynamic programming for Markov decision processes. *Mathematical Programming*, 125(2):235–261, 2010.
- [75] A. Ruszczyński and A. Shapiro. Stochastic programming models. *Handbooks in operations research and management science*, 10:1–64, 2003.
- [76] A. Ruszczyński and A. Shapiro. Optimization of convex risk functions. *Mathematics of Operations Research*, pages 433–452, 2006.
- [77] A. Ruszczyński and A. Shapiro. Optimization of risk measures. *Probabilistic and Randomized Methods for Design under Uncertainty*, pages 119–157, 2006.
- [78] A. Shapiro, D. Dentcheva, and A.P. Ruszczyński. *Lectures on stochastic programming: modeling and theory*, volume 9. Society for Industrial Mathematics, 2009.
- [79] A. Shapiro and A. Nemirovski. On complexity of stochastic programming problems. *Applied Optimization*, 99:111–146, 2005.
- [80] A. Shapiro, W. Tekaya, J. P. da Costa, and M.P. Soares. Report for technical cooperation between Georgia Institute of Technology and ONS. Technical report, Georgia Institute of Technology and ONS - Operador Nacional do Sistema Elétrico, 2011.
- [81] Alexander Shapiro. Monte Carlo sampling methods. In A. Ruszczyński and A. Shapiro, editors, *Stochastic Programming*, volume 10 of *Handbooks in Operations Research and Management Science*, pages 353 – 425. Elsevier, 2003.
- [82] B. W. Silverman. *Density Estimation for Statistics and Data Analysis*. Chapman & HALL/CRC, 1998.
- [83] J. S. Simonoff. *Smoothing methods in statistics*. Springer Verlag, 1996.
- [84] H.J. Sussmann. A maximum principle for hybrid optimal control problems. In *Proceedings of the 38th IEEE Conference on Decision and Control*, volume 1, pages 425 –430 vol.1, 1999.
- [85] F. Svahn. In-car navigation usage: An end-user survey on existing systems. *IRIS27, Falkenberg, Sweden*, 2004.
- [86] Office of Transportation US Environment Protection Agency and Air Quality.
- [87] P. Varaiya. Smart cars on smart roads: Problems of control. *IEEE Transactions On Automatic Control*, 38:195–207, 1993.
- [88] B. Verweij, S. Ahmed, A.J. Kleywegt, G. Nemhauser, and A. Shapiro. The sample average approximation method applied to stochastic routing problems: a computational study. *Computational Optimization and Applications*, 24:289–333, 2003.
- [89] L. Yu, J.-P. Barbot, D. Boutat, and D. Benmerzouk. Observability normal forms for a class of switched systems with Zeno phenomena. In *in Proceedings of the 2009 American Control Conference*, pages 1766–1771, 2009.
- [90] H. Zhang and M. R. James. Optimal control of hybrid systems and a systems of quasi-variational inequalities. *SIAM Journal of Control Optimization*, November 2005.
- [91] C. Zhihao. *LNG portfolio optimization*. PhD thesis, École Polytechnique, 2011.
- [92] S. Zoroofi. Modeling and simulation of vehicular power systems, 2008.

## Liste de Publications

### Journal avec comité de lecture

- Granato, G. and Zidani, H. *Level-set approach for reachability analysis of hybrid systems under lag constraints*, SIAM J. Control and Optimization, submitted, awaiting review.
- Granato, G. *Reachability of Hybrid Systems Using Level-set Methods*, Optimization and Engineering, submitted, awaiting review.

### Conférences avec actes

- Granato G. *Range Optimization of Hybrid Vehicles*, Proceedings of the International Conference of Engineering Optimization - ENG OPT, 2012, Rio de Janeiro, Brazil, <http://www.engopt.org/paper/297.pdf>.
- Aouchiche K., Bonnans F., Granato, G., and Zidani H. *A stochastic dynamic principle for hybrid systems with execution delay and decision lags*, Decision and Control and European Control Conference (CDC-ECC), 2011 50th IEEE Conference on , pp.6788-6793, 12-15 Dec. 2011, doi: 10.1109/CDC.2011.6161303.

### Communications internationales

- Granato G. and Zidani H. *HJB Equations for Non-Autonomous Hybrid Dynamical Systems*, Summer School and Workshop of the Initial Training Network for Sensitivity Analysis and Deterministic Optimal Control - SADCO, 2011, London, England, *invited speaker*.
- Aouchiche K., Bonnans F., Granato G. and Zidani H. *Stochastic Optimization of Hybrid Systems with Activation Delay and Decision Lag*, International Conference on Continuous Optimization- ICCOPT, 2010, Santiago, Chile, *oral communication*.

### Brevets déposés auprès de l'INPI

- Bonnans F., Granato G. *Procédé de génération de modèles de consommation d'énergie électrique adaptés au style de conduite du conducteur et aux spécificités du trajet*. Premier dépôt 02/12/2011.
- Aouchiche K., Bonnans F., Granato G., Rousseau, G. and Zidani H. *Loi de gestion d'énergie stochastique pour un véhicule hybride ou un véhicule électrique équipé d'un prolongateur d'autonomie avec contrôle autonome des arrêts/marches du moteur thermique considérant un délai de décision et un retard d'exécution*. Premier dépôt 14/12/2011.
- Aouchiche K., Granato G. and Zidani H. *Procédé de détermination de l'autonomie maximale d'un véhicule avec plus d'une source d'énergie*. Premier dépôt 15/03/2012.
- Granato G. and Zidani H. *Procédé de détermination des états énergétiques atteignables d'un véhicule avec plusieurs sources d'énergie*. Premier dépôt 15/03/2012.

