NNT : 2016SACLY019

THÈSE DE DOCTORAT
DE L'UNIVERSITE PARIS-SACLAY

préparée à

L'ENSTA ParisTech

ÉCOLE DOCTORALE N°574

École Doctorale de Mathématiques Hadamard (EDMH)

Spécialité de doctorat : Mathématiques Appliquées

par

Achille SASSI

Numerical methods for hybrid control and chance-constrained optimization problems

Thèse présentée et soutenue à Palaiseau, le 27 Janvier 2017

Composition du jury :

M. Emmanuel TRÉLAT	Président	Universtié Pierre et Marie Curie
M. Denis ARZELIER	Rapporteur	LAAS-CNRS
M. Alexander VLADIMIRSKY	Rapporteur	Cornell University
M. Pierre CARPENTIER	Examinateur	ENSTA ParisTech
Mme Hasnaa ZIDANI	Directrice de thèse	ENSTA ParisTech
M. Jean-Baptiste CAILLAU	Co-directeur de thèse	Université de Bourgogne
M. Max CERF	Co-directeur de thèse	Airbus Safran Launchers
M. Emmanuel TRÉLAT	Co-directeur de thèse	Universtié Pierre et Marie Curie

Science is whatever we want it to be.

Dr. Leo Spaceman

Acknowledgments

Among all the people who contributed to the realization of this work, i would like to thank first my director Hasnaa Zidani, who guided and helped me during one of the most important periods of my life.

I'm sincerely grateful to Jean-Baptiste Caillau, the co-director of my research. His support, suggestions and critiques revealed fundamental to the quality of the results we managed to obtain.

A special thank goes to Max Cerf, who provided the engineering insight and expertise from Airbus Safran Launchers. He was an inestimable resource of information for the correct interpretation of the abstract mathematical solutions.

Emmanuel Trélat is also among the people I couldn't thank enough. During our frequent meetings he was always prolific with creative approaches and ideas on how to tackle upcoming issues. His priceless contribution made this thesis possible.

Huge thanks also to both the referees of this thesis, Denis Arzelier and Alexander Vladimirsky, for helping me improve and correct my work with priceless suggestions and comments.

I want to express my eternal gratitude to Roberto Ferretti, my former director during my M.Sc. in Mathematics in Rome. He was the one who suggested me to pursuit a Ph.D. in the first place.

I also have to thank all the members of the SADCO Initial Training Network, as well as the colleagues at Unité de Mathématiques Appliquées (UMA) of ENSTA Paristech, in particular Pierre Carpentier, whose suggestions helped me improve some of the results this thesis.

Finally, a huge thank you to all the relatives, friends and electronic devices who were close to me during these difficult but life-changing three years. Especially my best friend Andrea and my PlayStation 4®: life has to try harder to separate us.

Contents

1 Introduction **1**

1.1 Part I: Numerical schemes for hybrid control systems 4

1.1.1 Contents 4

1.1.2 Structure 7

1.2 Part II: Chance-constrained optimization in aerospace 8

1.2.1 Contents 8

1.2.2 Structure 10

I Numerical schemes for hybrid control systems **11**

2 The hybrid optimal control problem **13**

2.1 Introduction 13

2.2 Preliminaries 14

2.2.1 Formulation of the problem 15

2.2.2 Basic assumptions 16

2.2.3 An example of hybrid control problem 18

2.2.4 Characterization of the value function 20

3 Approximation of the Hamilton-Jacobi-Bellman equation **23**

3.1 The numerical scheme 23

3.2 Policy Iteration and Semi-Lagrangian schemes 24

3.3 A Semi-Lagrangian scheme for hybrid control problems . . . 25

3.3.1 Numerical approximation 25

3.4 Policy Iteration algorithm 28

3.4.1 Modified policy iteration 33

3.5 Numerical tests 34

3.5.1 Stabilization of an unstable system 35

3.5.2 Three-gear vehicle 36

3.5.3 Bang–Bang control of a chemotherapy model 38

3.5.4 DC/AC inverter 40

4 Error estimates for the numerical scheme **45**
4.1 Cascade Problems . . . 46
4.1.1 Cascade for the HJB equation . . . 46
4.1.2 Cascade for the numerical scheme . . . 48
4.2 Lipschitz continuity . . . 49
4.3 Error estimates . . . 51
4.3.1 The Hamilton-Jacobi equation with obstacles . . . 51
4.3.2 Error estimates for the case without controlled jumps 56
4.3.3 The error estimate for the problem with n switches . . 59

A Appendix to Chapter 4 **63**
A.1 The upper bounds of the Lipschitz constants . . . 63
A.2 Lipschitz stability for the SL scheme . . . 64
A.3 Estimate on the perturbed value function of the stopping problem . . . 66
A.3.1 Estimate for the hitting times . . . 68
A.3.2 Estimate for the cost functionals . . . 72
A.3.3 Estimate for the value functions . . . 75
A.3.4 Estimate on the perturbed numerical approximation . 76

5 Conclusions **81**

II Chance-constrained optimization in aerospace **83**

6 The chance-constrained optimization problem **85**
6.1 Introduction . . . 85
6.1.1 The chance-constrained optimization problem . . . 88
6.2 Theoretical results . . . 92

7 Approximation of chance-constrained problems **97**
7.1 Numerical approaches . . . 97
7.1.1 Stochastic Arrow-Hurwicz Algorithm . . . 98
7.1.2 Kernel Density Estimation . . . 101
7.2 Numerical results . . . 103
7.2.1 Nonlinear optimization solvers . . . 104
7.2.2 Test 1: Simple single stage launcher with one decision variable and one random variable . . . 104
7.2.3 Test 2: Simple three stage launcher with three decision variables and three random variables . . . 117
7.2.4 Test 3: Simple three stage launcher with three decision variables and nine random variables . . . 126
7.2.5 Test 4: Simple single stage launcher with continuous control and one random variable . . . 131

7.2.6 Test 5: Goddard problem with one random variable . 138
7.2.7 Test 6: Complex three stage launcher with one decision variable and two random variables 145

8 Conclusions **161**

Chapter 1

Introduction

This Ph.D. thesis is the result of the collaboration between ENSTA ParisTech and Airbus Safran Launchers. This work is partially supported by the EU under the 7th Framework Programme Marie Curie Initial Training Network "FP7-PEOPLE-2010-ITN", SADCO project, GA number 264735-SADCO; and by iCODE Institute project funded by the IDEX Paris-Saclay, ANR-11-IDEX-0003-02.

The goal of this research project is the study of both hybrid systems and robust control problems in the domain of Applied Mathematics. Progress in these fields is crucial to the solution to problems arising in the industrial field. Thus the incentive to collaborate with the industrial sector.

In the field of engineering, and in our case aerospace engineering, solving optimization problems is a crucial aspect of every project. Whether it involves reducing the cost of a mission, finding the best trajectory for delivering a satellite to orbit or maximizing the payload of a space launcher. Optimal control problems are a particular type of optimization problems: they involve the study of the relation between the output and the input of a controlled system, sometimes with the additional goal of minimizing a cost associated to its evolution. The development of Control Theory is often driven by the great number and variety of its applications, such as the control of an engine, the management of a stock portfolio, or the organization of a power plant, to name a few.

A *controlled system* usually consists in a set of ordinary differential equations parameterized by a function called *control*:

$$\begin{cases} \dot{y}(t) = f\big(t, y(t), u(t)\big) \quad \forall t \in (0, t_{\mathrm{f}}] \\ y(0) = y_0 \end{cases}$$

where $f : \mathbb{R}_+ \times \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^n$ is the *state* function, $u : \mathbb{R}^m \to U \subset \mathbb{R}^m$ is the *control function*. In order to simplify the exposition, we assume that the final time t_{f} is finite. A control u is said to be *admissible* if it belongs to a given set $\mathcal{U}$, and each u in the admissible set selects a *trajectory* $y_u : \mathbb{R}_+ \to \mathbb{R}^n$. What

characterizes an optimal control problem is the presence of a *cost functional*, whose purpose is to measure the quality of a control strategy with respect to a chosen criterion. For a controlled system in the form presented, the cost functional $J : \mathcal{U} \to \mathbb{R}_+$ is defined as

$$J(u) := \phi(t_\mathrm{f}, y_u(t_\mathrm{f})) + \int_0^{t_\mathrm{f}} \ell(t, y_u(t), u(t))dt$$

where the functionals $\phi : \mathbb{R}_+ \times \mathbb{R}^n \to \mathbb{R}_+$ and $\ell : \mathbb{R}_+ \times \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}_+$ are called *final cost* and *running cost* respectively. Since we are interested in finding an *optimal control strategy* that minimizes the cost, we define our *optimal control problem* as

$$\min_{u \in \mathcal{U}} J(u).$$

There exist mainly two approaches to the solution of an optimal control problem: the Dynamic Programming Principle (DPP) and the Pontryagin Maximum Principle (PMP).

The Dynamic Programming Principle is used to associate the solution of a given optimization problem to a function of the system's initial state. If we explicit the dependency of the cost J on the initial state y_0 of the system, we can define a new function, called *value function*, as

$$v(y_0) := \inf_{u \in \mathcal{U}} J(y_0, u).$$

The DPP is then used to prove that, for every $0 < s < s_0$

$$v(y_0) = \inf_{u \in \mathcal{U}} \left\{ \int_0^s \ell(t, y(t), u(t))dt + v(y(s)) \right\}$$

where the parameter $\lambda > 0$ is called *discount factor*. The previous equation is, in fact, the mathematical translation of the *Principle of Optimality* stated by Bellman in his book [8]: "An optimal policy has the property that whatever the initial state and initial decision are, the remaining decisions must constitute an optimal policy with regard to the state resulting from the first decision". Using this property, the value function v can be characterized as the solution of a specific equation named *Dynamic Programming Equation* or *Bellman Equation*. The main advantage of this approach is that, once the value function is known, it's easy to reconstruct the solution to the original problem for any given initial state. However, this method has also a major drawback: the difficulty of the computation of this function increases dramatically with the dimension of the state space, a behavior appropriately named *Curse of Dimensionality*.

The Pontryagin Maximum Principle provides, for a given initial state, a condition for the optimal solution of a control problem. It differentiates from the DPP mainly because of the local nature of the optimality condition it provides, as opposed to the global one obtainable via the latter. Although

not particularly vulnerable to the increase of the problem's dimension, this technique does require a new computation of the solution if the initial state is changed. One of the advantages of the PMP is that it opens the path to the parameterization approach, leading to the *shooting method*. More precisely, the PMP allows to reformulate an optimal control problem as an ODE system in the form

$$\begin{cases} \dot{z}(t) = F(t, z(t)) & \forall t \in (0, t_\mathrm{f}) \\ R(z(0), z(t_\mathrm{f})) = 0. \end{cases}$$

Denoting $z_{z_0}(t_\mathrm{f})$ the solution of

$$\begin{cases} \dot{z}(t) = F(t, z(t)) & \forall t \in (0, t_\mathrm{f}) \\ z(0) = z_0 \end{cases}$$

our problem reduces to finding the root of the function $R(z_0, z_{z_0}(t_\mathrm{f}))$, which can be achieved numerically via iterative root-finding algorithms such as the Newton method. Unfortunately the PMP has also some drawbacks. The application of this principle requires some knowledge of the structure of the optimal control strategy, such as the presence of singular arcs or discontinuities, and in some cases these information might not be possible to recover. Moreover, the shooting method can have some stability issues in the presence of nonlinearities in the function F, thus requiring a good initial guess for z_0 in order to converge.

Regardless of the approach chosen, the computation of the numerical solution of an optimization problem can be achieved in several ways. For simpler problems, the solution method can be coded directly in programming languages such as C, C++, Fortran or Python or scripting languages like MATLAB, Mathematica or AMPL. For more complex problems though, it might be more convenient to use an already existing Non-Linear Problem Solver (NLP) as Ipopt, WORHP or KNITRO, which can be interfaced with many of the previously listed languages. In particular, the AMPL language has been designed to be easily paired with most of the available solvers, streamlining the task of simulating optimization problems.

The study presented in this work is focused on two particular aspects of optimization and optimal control: the numerical solution of a special case of optimal control problems, called *hybrid control problems*; and the approximation of chance-constrained optimization problems, with a focus on aerospace applications. Given the nature of these two different, although related, subjects, we decided to split this thesis into two parts.

1.1 Part I: Numerical schemes for hybrid control systems

This part is dedicated to the numerical approximation of the equations defining optimal control problems for hybrid systems as well as their algorithmic implementation.

The motivation behind this study is due to the interest that the author developed for the field of hybrid optimal control during the draft of his M.Sc. thesis in Italy. This, combined with the wide range of industrial applications of this theory, is a huge incentive to research and produce new results in the field.

1.1.1 Contents

Hybrid systems are described by a combination of continuous and discrete or logical variables and have been the subject of much attention over the last decade. One example of an optimal control problem involving a hybrid system arises in multi-stage rocket control, where there are both continuous control variables such as actuator inputs and logical controls governing the rocket stage ejection strategy and switches in the structure of the dynamic description.

More formally, consider the controlled system:

$$\begin{cases} \dot{X}(t) = f(X(t), Q(t), u(t)) \\ X(0) = x \\ Q(0^+) = q \end{cases}$$

where $x \in \mathbb{R}^d$, and q belongs to a finite set $\mathbb{I}$. Here, X and Q represent respectively the continuous and the discrete component of the state. The function f is the continuous dynamics and the continuous control set is $\mathcal{U}$. The trajectory undergoes discrete transitions when it enters two predefined sets $\mathcal{A}$ (the *autonomous* jump set) and $\mathcal{C}$ (the *controlled* jump set), both subsets of $\mathbb{R}^d \times \mathbb{I}$. More precisely:

- On hitting $\mathcal{A}$, the trajectory jumps to a predefined destination set $\mathcal{D}$ which is a subset of $\mathbb{R}^d \times \mathbb{I} \setminus (\mathcal{A} \cup \mathcal{C})$. This jump is guided by a prescribed transition map g and the arrival point after the jump depends on a discrete control action $\nu \in \mathcal{V}$.

- When the trajectory evolves in the set $\mathcal{C}$, the controller can choose to jump or not. If it chooses to jump, then the continuous trajectory displaced to a new point in $\mathcal{D}$.

The discrete inputs take place at the transition times

$$0 \leq \tau_0 \leq \tau_1 \leq \cdots \leq \tau_i \leq \tau_{i+1} \leq \dots$$
$$0 \leq \xi_0 \leq \xi_1 \leq \cdots \leq \xi_k \leq \xi_{k+1} \leq \dots$$

where τ_i denotes the time of each mandatory transition and ξ_k denotes the controlled transition times.

A hybrid control strategy θ would consist in the history of the control u and the transitions. For every initial state (x, q) and control strategy θ, we define the cost functional:

$$\begin{aligned} J(x,q;\theta) := &\int_0^{+\infty} \ell(X(t),Q(t),u(t))e^{-\lambda t}dt + \\ &+ \sum_{i=0}^{\infty} c_{\mathcal{A}}(X(\tau_i^-),Q(\tau_i^-),\nu_i)e^{-\lambda\tau_i} + \\ &+ \sum_{k=0}^{\infty} c_{\mathcal{C}}(X(\xi_k^-),Q(\xi_k^-),X(\xi_k^+),Q(\xi_k^+))e^{-\lambda\xi_k} \end{aligned}$$

where $\lambda > 0$ is the discount factor, ℓ is the running cost, $c_{\mathcal{A}}$ is the autonomous transition cost and $c_{\mathcal{C}}$ is the controlled transition cost.

With these definitions, the hybrid optimal control problem can now be stated as the computation of the control strategy θ^* which minimizes the cost J over the set Θ of all admissible strategies:

$$\inf_{\theta\in\Theta} J(x,q;\theta).$$

We solve this problem by applying the Dynamic Programming Principle, thus associating the solution of a the optimization problem to a function of the system's initial state. This so called value function is characterized as the solution of a specific equation named Dynamic Programming Equation. Once the value function is known, it's easy to reconstruct the solution to the original problem for any given initial state. However, the difficulty of the computation of this function increases dramatically with the problem's dimension d.

In our case, the value function V is defined as

$$V(x,q) := \inf_{\theta\in\Theta} J(x,q;\theta)$$

and, under suitable hypotheses, it can be proven that it satisfies

$$\begin{cases} \lambda V(x,q) + H(x,q,D_xV(x,q)) = 0 & \text{on } (\mathbb{R}^d \times \mathbb{I}) \setminus (\mathcal{A}\cup\mathcal{C}) \\ \max\{\lambda V(x,q) + H(x,q,D_xV(x,q)), \\ \qquad V(x,q) - \mathcal{N}V(x,q)\} = 0 & \text{on } \mathcal{C} \\ V(x,q) - \mathcal{M}V(x,q) = 0 & \text{on } \mathcal{A} \end{cases}$$

where H is the Hamiltonian, defined as

$$H(x,q,p) := \sup_{u\in U} \{ -\ell(x,q,u) - f(x,q,u)\cdot p\}$$

and $\mathcal{M}$ and $\mathcal{N}$ are the transition operators:

$$\mathcal{M}\phi(x,q) := \inf_{\nu\in\mathcal{V}} \Big\{\phi(g(x,q,\nu)) + c_{\mathcal{A}}(x,q,\nu)\Big\} \quad (x,q)\in\mathcal{A}$$
$$\mathcal{N}\phi(x,q) := \inf_{(x',q')\in\mathcal{D}} \{\phi(x',q') + c_{\mathcal{C}}(x,q,x',q')\} \quad (x,q)\in\mathcal{C}$$

For the approximation of this problem we can use a numerical scheme in the general form

$$\begin{cases} S(h,x,q,V_h(x,q),V_h) = 0 & \text{on } (\mathbb{R}^d\times\mathbb{I})\setminus(\mathcal{A}\cup\mathcal{C}) \\ \max\Big\{S(h,x,V_h(x,q),V_h), \\ \qquad V_h(x,q) - \mathcal{N}V_h(x,q)\Big\} = 0 & \text{on } \mathcal{C} \\ V_h(x,q) - \mathcal{M}V_h(x,q) = 0 & \text{on } \mathcal{A} \end{cases}$$

where V_h denotes the numerical approximation of the value function V, and $V_h(x,q)$ its value at the point (x,q).

The first contribution of this thesis in the field of hybrid optimal control is the derivation of an important error estimate between the value function V and its approximation V_h. We describe in detail the technique used to overcome the issues arising from the presence of the highly non-linear operators $\mathcal{M}$ and $\mathcal{N}$.

The second contribution is the adaptation of the Policy Iteration algorithm to the hybrid case in order to accelerate the convergence of the scheme S.

Usually, an approximation scheme is constructed over a discrete grid of nodes in (x_i,q) with discretization parameters Δx and Δt. We denote the discretization steps in compact form by $h := (\Delta t, \Delta x)$ and the approximate value function by V_h.

The scheme S can be rewritten write in the fixed point form

$$V_h(x_i,q) = \begin{cases} M^h V_h(x_i,q) & (x_i,q)\in\mathcal{A} \\ \min\{\Sigma^h(x_i,q,V_h), N^h V_h(x_i,q)\} & (x_i,q)\in\mathcal{C} \\ \Sigma^h(x_i,q,V_h) & \text{else} \end{cases}$$

in which N^h, M^h and Σ^h are approximations for respectively the operators $\mathcal{N}$, $\mathcal{M}$ and the Hamiltonian H.

Under basic assumptions, the right-hand side of the scheme's fixed point form is a contraction and can therefore be solved by fixed-point iteration, also known as Value Iteration (VI):

$$V_h^{(j+1)} = T^h(V_h^{(j)}).$$

Since in practice this procedure might require a large number of iterations in order to converge, our objective is to provide a better alternative to VI.

The Policy Iteration (PI) algorithm is used to solve optimization problems in the form

$$\min_{y \in Y \subseteq \mathbb{R}^m} \{B(y)x - c(y)\} = 0$$

where, for every y, $B(y)$ is a $n \times n$ matrix and $c(y)$ is an n-dimensional array. The procedure consists in alternating two operations, called Policy Evaluation and Policy Improvement, in the following way:

$j \leftarrow 0$
STOP $\leftarrow$ FALSE
$y_0 \in Y$
while STOP = FALSE **do**
  if [stopping criterion satisfied] **then**
    STOP $\leftarrow$ TRUE
  else
    $x_j \leftarrow \xi$ solution of $B(y_j)\xi = c(y_j)$ (*Policy Evaluation*)
    $(y_{j+1}) \leftarrow \arg\min_{\tau \in Y} \{B(\tau)x_j - c(\tau)\}$ (*Policy Improvement*)
    $j \leftarrow j + 1$
  end if
end while

We point out that this algorithm might need to be coupled with some selection criteria in order to properly define the Policy Improvement step.

In this part show how to construct a scheme S that can be solved via PI and use several numerical test to show that this techniques is indeed very efficient in practice when compared to VI.

In the following sections we also obtain the error estimates between the value function V and its approximation V_h. We study the simplified hybrid optimal control problem in which optional transitions are always allowed and mandatory transitions are always forbidden, meaning that $\mathcal{C} = \mathbb{R}^d \times \mathbb{I}$ and $\mathcal{A} = \emptyset$. In this framework, we are able to prove our main result:

$$-\underline{C} \ln h |h|^\gamma \leq V(x,q) - V_h(x,q) \leq \overline{C} \ln h |h|^\gamma \quad \forall (x,q) \in \mathbb{R}^d \times \mathbb{I}$$

for some positive constants $\underline{C}$, $\overline{C}$ and γ, defined explicitly.

1.1.2 Structure

In Chapter 2 we give some historical notes and a formal definition of hybrid optimal control problems, describing the mathematical framework and providing the analytical background.

Chapter 3 analyzes the approximation of such problems and shows a comparison between the numerical solution obtained by using two implementations of the Semi-Lagrangian scheme: Value Iteration and Policy Iteration.

Chapter 4 and Appendix A complement the previous chapter with theoretical results on the error estimates of numerical schemes for hybrid optimal control problems, providing convergence and regularity results on the approximated solution.

Finally, Chapter 5 is dedicated to conclusions.

1.2 Part II: Chance-constrained optimization in aerospace

This part is dedicated instead to the chance-constrained approach for the solution of robust optimization problems, focusing on the application of the Kernel Density Estimation technique.

Robust methods are aimed at achieving consistent performance and/or stability in the presence of bounded modeling errors. Using again the case of a multi-stage rocket, an example of robust optimization would consist in minimizing the initial fuel load of the launcher in the presence of uncertainties in the engine thrust, while guaranteeing that the payload is delivered within a certain level of confidence. One of the many possible approaches used for solving robust optimization problems consists in chance-constrained optimization. The name comes from the the idea of treating the uncertainties in the underlying mathematical model as random variables.

1.2.1 Contents

Consider the problem of minimizing a function J which depends on a set of variables $x \in \mathcal{X} \subseteq \mathbb{R}^n$, while requiring a given constraint G to be satisfied:

$$\begin{cases} \min\limits_{x \in \mathcal{X}} J(x) \\ G(x) \geq 0. \end{cases}$$

Let us suppose that, in addition to the decision variable x, the constraint function G also depends on a set of parameters $\xi \in \mathbb{R}^m$. This means that to every choice of ξ corresponds one instance of the problem:

$$\begin{cases} \min\limits_{x \in \mathcal{X}} J(x) \\ G(x, \xi) \geq 0. \end{cases}$$

At this point we might be interested in studying the dependency of the solution to our parameterized problem on ξ, or require the inequality constraint to be satisfied for a subset $\mathcal{E} \subseteq \mathbb{R}^m$ of all the possible values of ξ, for example:

$$\begin{cases} \min\limits_{x \in \mathcal{X}} J(x) \\ G(x, \xi) \geq 0 \quad \forall \xi \in \mathcal{E}. \end{cases}$$

A chance-constrained optimization problem arises from the assumption that the components of ξ are random variables with a given distribution. This allows us to reformulate our optimization problem in the form:

$$\begin{cases} \min\limits_{x \in \mathcal{X}} J(x) \\ \mathbb{P}[G(x, \xi) \geq 0] \geq p. \end{cases}$$

where $p \in (0, 1)$ is a probability threshold and ξ is an m-dimensional random vector defined on some probability space.

Optimization problems with chance constraints are often considered if there's a need of minimizing a cost associated to the performance of a dynamical model, while taking into account uncertainties in the parameters defining it. Many results in this field are related to the theoretical study of this type of problems, such as the regularity of the constraint function and the stability of the solution with respect to the distribution of ξ.

Our main contribution focuses on the study of an efficient numerical solution to chance-constrained problems. In particular, we explore the application of the Kernel Density Estimation (KDE): a technique used in non-parametric statistics to approximate the probability density function of a random variable with unknown distribution.

The main difficulty lies in the form of the constraint function: being G dependent on both x and ξ, it is a priori impossible to derive an analytical representation of its probability distribution, even if the distribution of ξ is known.

Our idea consists in producing an approximation of the distribution of G, so that we can replace the probability with the integral of its estimated density and solve the stochastic optimization problem as a deterministic one. More precisely, for a given x in $\mathcal{X}$, let f_x and $\hat{f}_x$ be respectively the probability density function (pdf) of $G(x, \xi)$ and its approximation. From basic probability theory, we have

$$\mathbb{P}[G(x, \xi) \geq 0] = 1 - \mathbb{P}[G(x, \xi) < 0] = 1 - \int_{-\infty}^{0} f_x(z) dz$$

if f_x and $\hat{f}_x$ are "close" in some appropriate sense, we obtain

$$\int_{-\infty}^{0} \hat{f}_x(z) dz \approx \int_{-\infty}^{0} f_x(z) dz = 1 - \mathbb{P}[G(x, \xi) \geq 0].$$

By defining $\hat{F}_x(y) := \int_{-\infty}^{y} \hat{f}_x(z) dz$ we can write an approximation of our chance constraint in the form

$$\begin{cases} \min\limits_{x \in \mathcal{X}} J(x) \\ \hat{F}_x(0) \leq 1 - p. \end{cases}$$

Such an approximation can be obtained via KDE: for every $x \in \mathcal{X}$, let $\{G(x,\xi_1), G(x,\xi_2), \dots, G(x,\xi_n)\}$ be a sample of size n from the variable $G(x,\xi)$. A Kernel Density Estimator for the pdf f_x is the function

$$\hat{f}_x(y) := \frac{1}{nh} \sum_{i=1}^{n} K\left(\frac{y - G(x,\xi_i)}{h}\right)$$

where the function K is called kernel and the smoothing parameter h is called bandwidth.

We show how this method can be implemented numerically as a valid alternative to the traditional Monte Carlo approach, and use several examples of chance-constrained optimization in the domain of aerospace to test its performances.

1.2.2 Structure

In Chapter 6 we give the mathematical formulation of the chance-constrained optimization problem and provide an overview of the existing theoretical results.

Chapter 7 collects several numerical results on the application of two different techniques for the numerical solution of chance-constrained optimization problems: the Stochastic Arrow-Hurwicz Algorithm and Kernel Density Estimation. The former combines the Monte Carlo method with the Arrow-Hurwicz iterative gradient method. The latter is a technique used in non-parametric statistics for approximating the density function of a sample with unknown distribution.

Chapter 8 is dedicated to conclusions.

Part I

Numerical schemes for hybrid control systems

Chapter 2

The hybrid optimal control problem

2.1 Introduction

The earlier examples of the application of Control Theory date back to the end of the nineteenth century, but the most visible growth of this fringe discipline between Mathematics and Engineering happened during the Second World War. The field of application of Control Theory then extended from the development of military and aerospace technologies to more abstract problems, raising the interest of many mathematicians. It was in fact during the Space Race that the two powers involved in the Cold War gave birth to the traditional approaches for the solution of control and optimization problems.

In 1957 the American Richard E. Bellman illustrated, in his book called "Dynamic Programming", the well known *Dynamic Programming Principle* (a term he coined himself), which is used to associate the solution of a given optimization problem to a function of the system's initial state. This function is called *value function* and it's characterized as the solution of a specific equation named *Dynamic Programming Equation* or *Bellman Equation*. Once the value function is known, it's easy to reconstruct the solution to the original problem for any given initial state.

In the following years, the growing interest in the application of control problems required the introduction of a new notion of solution for partial differential equations which don't admit one in the classical sense. In the eighties, mathematicians Pierre-Louis Lions and Michael G. Crandall developed the concept of *viscosity solution*, through which it's possible to derive the uniqueness of the solution to some differential equations related to a great variety of optimal control problems.

The last years of the twentieth century saw another expansion of the field of Control Theory with its generalization to *hybrid systems*, that is, complex

systems in which continuous and discrete control actions mix together.

In this chapter we set the basic assumptions on the hybrid control problem and review the characterization of the value function in terms of a suitable Dynamic Programming equation.

2.2 Preliminaries

We start by introducing some notations. We denote $|\cdot|$ the standard Euclidean norm in any $\mathbb{R}^n$ type space (for any $n \geq 1$). In particular, if B is a $n \times n$ matrix, then $|B|^2 = tr(BB^\intercal)$, where $B^\intercal$ is the transpose of B and $|B|$ is the Frobenius norm. For a discrete set S, $|S|$ will denote its cardinality.

Let ϕ be a bounded function from $\mathbb{R}^n$ into either $\mathbb{R}$, $\mathbb{R}^n$, or the space of $n \times m$ matrices ($m \geq 1$). We define

$$|\phi|_0 := \sup_{x \in \mathbb{R}^n} |\phi(x)|.$$

If ϕ is also Lipschitz continuous, we set

$$|\phi|_1 := \sup_{x,y \in \mathbb{R}^n, x \neq y} \frac{|\phi(x) - \phi(y)|}{|x - y|}$$

Moreover, for any closed set $\mathcal{S} \subset \mathbb{R}^n$, the space $C_{\mathrm{b}}(\mathcal{S})$ [respectively $C_{\mathrm{b,l}}(\mathcal{S})$] will denote the space of continuous and bounded functions [resp. bounded and Lipschitz continuous functions] from $\mathcal{S}$ to $\mathbb{R}$.

Given $\phi \in \left[C_{\mathrm{b,l}}(\mathbb{R}^n)\right]^m$, we denote by L_ϕ and M_ϕ some *upper bounds* of respectively the Lipschitz constant and the supremum of ϕ:

$$\begin{aligned} L_\phi &\geq \max_{i \in \{1,\dots,m\}} |\phi_i|_1 \\ M_\phi &\geq |\phi|_0. \end{aligned}$$

We denote by $\leq$ the component wise ordering in $\mathbb{R}^n$, and by $\preceq$ the ordering in the sense of positive semi-definite matrices. For any $a, b \in \mathbb{R}$, we define $a \wedge b$ as

$$a \wedge b := \min(a, b).$$

For any given closed subset $\mathcal{S}$ of $\mathbb{R}^m$, the notations $\partial\mathcal{S}$, $\mathrm{dist}(\cdot, \mathcal{S})$ stand respectively for the boundary of $\mathcal{S}$ and the Euclidean distance defined by

$$\mathrm{dist}(x, \mathcal{S}) := \inf_{y \in \mathcal{S}} |x - y|.$$

Let us also recall that in the inductive limit topology on $\mathbb{R}^d \times \mathbb{I}$, the concept of converging sequence is stated as follows.

Definition 2.2.1 (convergence of a sequence)**.** $(x_n, q_n) \in \mathbb{R}^d \times \mathbb{I}$ *converges to* $(x, q) \in \mathbb{R}^d \times \mathbb{I}$ *if, for any* $\varepsilon > 0$*, there exists* N_ε *such that* $q_n = q$*, and* $|x_n - x| < \varepsilon$ *for any* $n \geq N_\varepsilon$*.*

2.2.1 Formulation of the problem

Among the various mathematical formulations of optimal control problems for hybrid systems, we will adopt here the one given in [26, 15, 4]. Let therefore $\mathbb{I}$ be a finite set, and consider the controlled system (X, Q) satisfying:

$$\begin{cases} \dot{X}(t) = f(X(t), Q(t), u(t)) \\ X(0) = x \\ Q(0^+) = q \end{cases} \tag{2.2.1}$$

where $x \in \mathbb{R}^d$, and $q \in \mathbb{I}$. Here, X and Q represent respectively the continuous and the discrete components of the state. Throughout the thesis we term *switch* a transition in the state which involves only a change in the $Q(t)$ component, whereas *jump* denotes a transition which might also involve a discontinuous change in $X(t)$.

The function $f : \mathbb{R}^d \times \mathbb{I} \times \mathcal{U} \to \mathbb{R}^d$ is the continuous dynamics and the continuous control set is:

$$\mathcal{U} = \{u : (0, +\infty) \to U \quad | \quad u \text{ measurable, } U \text{ compact metric space}\}.$$

In our case, the function u represents a feedback control law: u adapts dynamically as the system changes through time, steering its evolution towards the desired goal. This type of control is also called *closed-loop control*, since changes in the system affect the control strategy, which in turn affects the system.

The trajectory undergoes discrete transitions when it enters two predefined sets $\mathcal{A}$ (the *autonomous* jump set) and $\mathcal{C}$ (the *controlled* jump set), both of them subsets of $\mathbb{R}^d \times \mathbb{I}$. More precisely:

- On hitting $\mathcal{A}$, the trajectory jumps to a predefined destination set $\mathcal{D}$, possibly with a different discrete state $q' \in \mathbb{I}$. This jump is guided by a prescribed transition map $g : \mathbb{R}^d \times \mathbb{I} \times \mathcal{V} \to \mathcal{D}$, where $\mathcal{V}$ is a discrete finite control set. We denote by τ_i an arrival time to $\mathcal{A}$, and by $(X(\tau_i^-), Q(\tau_i^-))$ the position of the state before the jump. The arrival point after the jump and the new discrete state value will be denoted by $(X(\tau_i^+), Q(\tau_i^+)) = g(X(\tau_i^-), Q(\tau_i^-), \nu_i)$ and will depend on a discrete control action $\nu_i \in \mathcal{V}$.

- When the trajectory evolves in the set $\mathcal{C}$, the controller can choose to jump or not. If it chooses to jump, then the continuous trajectory is displaced to a new point in $\mathcal{D}$. By ξ_i we denote a (controlled) transition time. The state $(X(\xi_i^-), Q(\xi_i^-))$ is moved by the controlled jump the to the destination $(X(\xi_i^+), Q(\xi_i^+)) \in \mathcal{D}$.

The trajectory starting from $x \in \mathbb{R}^d$ with discrete state $q \in \mathbb{I}$ is therefore composed of a continuous evolution given by (2.2.1) between two discrete

jumps at the transition times. For example, assuming $\tau_i < \xi_k < \tau_{i+1}$, the evolution of the hybrid system would be given by:

$$\begin{cases} (X(\tau_i^+), Q(\tau_i^+)) = g(X(\tau_i^-), Q(\tau_i^-), \nu) & \\ \dot{X}(t) = f(X(t), Q(\tau_i^+), u(t)) & \tau_i < t < \xi_k \\ (X(\xi_k^+), Q(\xi_k^+)) \in \mathcal{D} & (\text{destination of the jump at } \xi_k) \\ \dot{X}(t) = f(X(t), Q(\xi_k^+), u(t)) & \xi_k < t < \tau_{i+1} \end{cases}$$

Associated to this hybrid system, we consider an infinte horizon control problem where the cost is composed of a running cost and transistion costs corresponding to the controlled and uncontrolled jumps. A similar control problem has been considered in [28], where the authors have studied the value function and its numerical approximation. A procedure to compute a piecewise constant feedback control is also analyzed in [28].

2.2.2 Basic assumptions

In the product space $\mathbb{R}^d \times \mathbb{I}$, we consider sets (and in particular the sets $\mathcal{A}, \mathcal{C}$ and $\mathcal{D}$) of the form

$$\mathcal{S} = \{(x, q) \in \mathbb{R}^d \times \mathbb{I} : x \in S_q\} \tag{2.2.2}$$

in which $\mathcal{S}_i$ represents the subset of $\mathcal{S}$ in which $q = i$.

We make the following standing assumptions on the sets $\mathcal{A}, \mathcal{C}, \mathcal{D}$ and on the functions f and g:

(A1) For each $i \in \mathbb{I}$, $\mathcal{A}_i$, $\mathcal{C}_i$, and $\mathcal{D}_i$ are closed subsets of $\mathbb{R}^d$, and $\mathcal{D}_i$ is bounded. The boundaries $\partial\mathcal{A}_i$ and $\partial\mathcal{C}_i$ are C^2.

This assumption is essential to the well-posedness of the HJB equation resulting from the characterization of the value function. The regularity of the boundary of $\mathcal{A}$ and $\mathcal{C}$ is also necessary for proving some stability results.

(A2) We assume that in the case $\mathcal{A}_i$, $\mathcal{C}_i$, and $\mathcal{D}_i$ are non-empty, for all $i \in \mathbb{I}$,

$$\operatorname{dist}(\mathcal{A}_i, \mathcal{C}_i) \geq \beta > 0 \quad \text{and} \quad \operatorname{dist}(\mathcal{A}_i, \mathcal{D}_i) \geq \beta > 0.$$

The purpose of the first inequality is to ensure that the intersection between $\mathcal{A}$ and $\mathcal{C}$ is empty, thus avoiding any ambiguity on the position of the state inside $\mathbb{R}^d \times \mathbb{I}$.

The second inequality, along with suitable assumptions on the cost functional, prevents any pathological executions of multiple discrete transitions at one single time or an infinite number of discrete transitions in any finite period of time. These kind of transitions happen because the definition of admissible controls implies that a discrete

transition on the state is instantaneous and takes no time, they are known in the literature as *Zeno executions*. See also [9, 72] for other kind of sufficient assumptions that allow to avoid Zeno executions.

(A3) The function f is Lipschitz continuous with Lipschitz constant L_f in the state variable x and uniformly continuous in the control variable u. Moreover, for all $(x, q) \in \mathbb{R}^d \times \mathbb{I}$ and $u \in U$,

$$|f(x, q, u)| \leq M_f$$

This hypothesis could be replaced by the less strict requirement of f being only locally Lipschitz, in order to extend the results to a more general case. However, the boundedness of f is a standard simplification in the framework of hybrid optimal control problems (see [3] and [4]).

(A4) The map $g : \mathcal{A} \times \mathcal{V} \to \mathcal{D}$ is bounded and uniformly Lipschitz continuous with respect to x.

(A5) $\partial\mathcal{A}$ is a compact set, and for some $\omega > 0$, the transversality condition

$$f(x, q, u) \cdot \eta_{x,q} \leq -2\omega$$

holds for all $x \in \partial\mathcal{A}_q$, and all $u \in U$, where $\eta_{x,q}$ denotes the unit outward normal to $\partial\mathcal{A}_q$ at x. We also assume similar transversality conditions on $\partial\mathcal{C}$.

This condition, also known as the *Petrov condition*, ensures that the boundary of the sets $\mathcal{A}$ and $\mathcal{C}$ are attractive with respect to the system dynamics. Informally, this forces the state of the system to enter $\mathcal{A}$ or $\mathcal{C}$ if it gets close enough to its boundary. Similar conditions are also essential to some controllability results such as the *small time controllability* analyzed in [3, Chapter 4]. In this book, the authors study optimal control problems with cost functionals involving the exit time from a given domain. The concept of small time controllability is introduced to study the continuity of the value function.

In what follows, a control policy for the hybrid system consists in two parts: continuous input u and discrete inputs. A continuous control is a measurable function $u \in \mathcal{U}$ acting on the trajectory through the continuous dynamics (2.2.1). The discrete inputs take place at transition times

$$0 \leq \tau_0 \leq \tau_1 \leq \cdots \leq \tau_i \leq \tau_{i+1} \leq \dots$$
$$0 \leq \xi_0 \leq \xi_1 \leq \cdots \leq \xi_k \leq \xi_{k+1} \leq \dots.$$

At time τ_i (which *cannot* be selected by the controller) the trajectory undergoes a discrete transition under the action of the discrete control $w_i \in \mathcal{V}$,

while at time ξ_k (which *can* be selected by the controller) the trajectory moves to a new position $(x'_k, q'_k) \in \mathcal{D}$. The discrete inputs are therefore of two forms $\{\nu_i\}_{i\geq 0}$ and $\{(\xi_k, x'_k, q'_k)\}_{k\geq 0}$. To shorten the notation, we will denote by $\theta := \Big(u(\cdot), \{\nu_i\}, \{(\xi_k, x'_k, q'_k)\}\Big)$ a hybrid control strategy, and by Θ the set of all admissible strategies.

Now, for every control strategy $\theta \in \Theta$, we associate the cost defined by:

$$\begin{aligned} J(x, q; \theta) := &\int_0^{+\infty} \ell(X(t), Q(t), u(t)) e^{-\lambda t} dt + \\ &+ \sum_{i=0}^{\infty} c_{\mathcal{A}}(X(\tau_i^-), Q(\tau_i^-), \nu_i) e^{-\lambda \tau_i} + \\ &+ \sum_{k=0}^{\infty} c_{\mathcal{C}}(X(\xi_k^-), Q(\xi_k^-), X(\xi_k^+), Q(\xi_k^+)) e^{-\lambda \xi_k} \end{aligned} \tag{2.2.3}$$

where $\lambda > 0$ is the discount factor, $\ell : \mathbb{R}^d \times \mathbb{I} \times U \to \mathbb{R}_+$ is the running cost, $c_{\mathcal{A}} : \mathcal{A} \times \mathcal{V} \to \mathbb{R}_+$ is the autonomous transition cost and $c_{\mathcal{C}} : \mathcal{C} \times \mathcal{D} \to \mathbb{R}_+$ is the controlled transition cost. The value function V is then defined as:

$$V(x, q) := \inf_{\theta \in \Theta} J(x, q; \theta). \tag{2.2.4}$$

We assume the following conditions on the cost functional:

(A6) $\ell : \mathbb{R}^d \times \mathbb{I} \times U \to \mathbb{R}$ is a bounded and nonnegative function, Lipschitz continuous with respect to the x variable, and uniformly continuous with respect to the u variable.

(A7) $c_{\mathcal{A}} : \mathcal{A} \times \mathcal{V} \to \mathbb{R}$ and $c_{\mathcal{C}} : \mathcal{C} \times \mathcal{D} \to \mathbb{R}$ are bounded with a strictly positive infimum $K_0 > 0$ and both $c_{\mathcal{A}}$ and $c_{\mathcal{C}}$ are uniformly Lipschitz continuous in the variable x'.

2.2.3 An example of hybrid control problem

For the purpose of showing how the framework of hybrid systems can be adapted to industrial applications, we provide a typical example of a hybrid control problem involving the simulation of a vehicle equipped with two engines: an electric engine (EE) and an internal combustion engine (ICE) (figure 2.2.1).

The former is powered by a battery that is recharged by the latter, which instead consumes regular fuel. The goal is to minimize a combination of fuel consumption and speed by acting on both the acceleration strategy and the discontinuous switching between EE and ICE. The commutation between the two engines is mandatory only in the cases of a fully charged or fully discharged battery: in the first case the vehicle has to switch from ICE to EE, while in the second case the opposite happens. For all the intermediate

level of battery charge, is up to the control strategy to transition from EE to ICE or vice versa.

Before giving the details of the model, we remark that this example, as well as all the others in this part, do not necessarily satisfy all the theoretical assumptions. The examples serve the purpose of illustrating the mathematical framework, explaining the application of the numerical methods and testing the necessity of the assumptions.

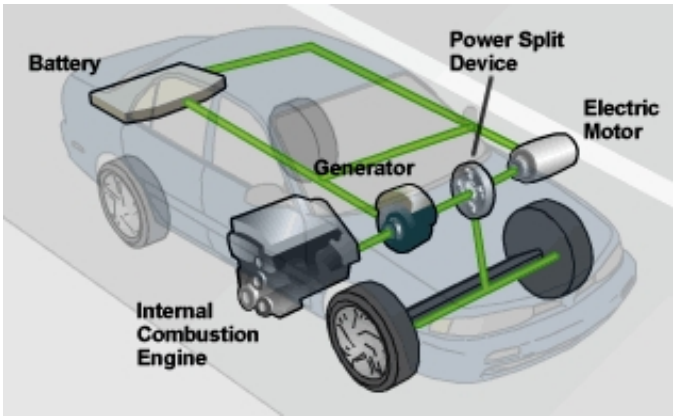


Figure 2.2.1: A hybrid car. Credits to the U.S. Department of Energy (`https://www.fueleconomy.gov/feg/hybridtech.shtml`).

The state variable $X(t)$ of the system is the normalized residual charge of the battery at time t, such that $X(t) \in [0,1]$ for every $t \in [0,+\infty)$. We control the vehicle by means of its speed and active engine. The speed $u(t)$ is also normalized: $u(t) \in [0,1]$ for every $t \in [0,+\infty)$. The active engine at time t is denoted by $Q(t)$:

$$Q(t) := \begin{cases} 1 & \text{EE is active} \\ 2 & \text{ICE is active} \end{cases}$$

The dynamics are described by the system of ordinary differential equations

$$\dot{X}(t) = f(X(t), Q(t), u(t)) := \begin{cases} -(1 - X(t))u(t) & Q(t) = 1 \\ u(t) & Q(t) = 2 \end{cases}$$

with initial conditions $X(0) = 0$ and $Q(0) = 2$. From the previous equations it can be deduced that the battery discharges at an increasing rate while the EE is active, while it charges linearly with respect to the control u if the ICE is active.

In this case we have four sets defining the switching conditions: $\mathcal{A}_1$, $\mathcal{C}_1$, $\mathcal{A}_2$, $\mathcal{C}_2$. Because it is mandatory to switch from EE to ICE if the battery level is zero, for the first dynamic we have $\mathcal{A}_1 := \{0\}$ and $\mathcal{C}_1 := (0,1]$. On

the opposite, for the second dynamic the sets are $\mathcal{A}_2 := \{1\}$ and $\mathcal{C}_2 := [0, 1)$. Moreover, instantaneous jumps in the state variable are obviously not allowed, since they would imply an unrealistic discontinuous behavior of the battery charge.

At any time, the objective is to minimize the function

$$\ell(X(t), Q(t), u(t)) = \begin{cases} -u(t) & Q(t) = 1 \\ (u(t) - 1)u(t) & Q(t) = 2 \end{cases}$$

which translates to the necessity of maximizing the speed of the vehicle while reducing fuel consumption. Finally, taking into account that turning on the ICE takes time and fuel, we assume that the switching cost from dynamic 1 to 2 is strictly positive, as opposed to a zero cost of switching from dynamic 2 to 1.

2.2.4 Characterization of the value function

We briefly review the main theoretical facts about the value function V defined in (2.2.4).

It is quite straightforward to derive the Dynamic Programming Principle for the control problem (2.2.1)–(2.2.3) as follows:

1. For any $(x, q) \in (\mathbb{R}^d \times \mathbb{I}) \setminus (\mathcal{A} \cup \mathcal{C})$ there exists $s_0 > 0$ such that, for every $0 < s < s_0$, we have:

$$V(x, q) = \inf_{u \in \mathcal{U}} \left\{ \int_0^s \ell(X(t), q, u(t)) e^{-\lambda t} dt + e^{-\lambda s} V(X(s), q) \right\} \quad (2.2.5)$$

2. For $(x, q) \in \mathcal{A}$, we have:

$$V(x, q) = \inf_{\nu \in \mathcal{V}} \Big\{ V(g(x, q, \nu)) + c_{\mathcal{A}}(x, q, \nu) \Big\} \quad (2.2.6)$$

3. For $(x, q) \in \mathcal{C}$, we have:

$$V(x, q) \leq \inf_{(x', q') \in \mathcal{D}} \{V(x', q') + c_{\mathcal{C}}(x, q, x', q')\} \quad (2.2.7)$$

   If it happens that $V(x, q) < \inf_{(x', q') \in \mathcal{D}} \{V(x', q') + c_{\mathcal{C}}(x, q, x', q')\}$, then there exists $s_0 > 0$ such that for every $0 < s < s_0$, we have:

$$V(x, q) = \inf_{u \in \mathcal{U}} \left\{ \int_0^s \ell(X(t), q, u(t)) e^{-\lambda t}\, dt + e^{-\lambda s} V(X(s), q) \right\} \quad (2.2.8)$$

Moreover, it is known that the value function V is uniformly continuous [26, Theorem 3.5]. More precisely, we have:

Lemma 2.2.1. *Under assumptions (A1)–(A7), the value function V is Hölder continuous and bounded.*

From the dynamic programming principle, it can be checked that the value function satisfies, in an appropriate sense, a quasi-variational inequality. To give a precise statement of this result, we first introduce the Hamiltonian $H : \mathbb{R}^d \times \mathbb{I} \times \mathbb{R}^d \to \mathbb{R}$ defined, for $x, p \in \mathbb{R}^d$ and $q \in \mathbb{I}$, by:

$$H(x,q,p) := \sup_{u \in U} \{ -\ell(x,q,u) - f(x,q,u) \cdot p \} \tag{2.2.9}$$

We also define the transition operators $\mathcal{M}$ [respectively $\mathcal{N}$] mapping $C^0(\mathbb{R}^d \times \mathbb{I})$ into $C^0(\mathcal{A})$ [resp. $C^0(\mathcal{C})$] by:

$$\mathcal{M}\phi(x,q) := \inf_{\nu \in \mathcal{V}} \Big\{ \phi(g(x,q,\nu)) + c_{\mathcal{A}}(x,q,\nu) \Big\} \quad (x,q) \in \mathcal{A} \tag{2.2.10}$$

$$\Big[\text{resp.} \quad \mathcal{N}\phi(x,q) := \inf_{(x',q') \in \mathcal{D}} \{ \phi(x',q') + c_{\mathcal{C}}(x,q,x',q') \} \quad (x,q) \in \mathcal{C} \Big] \tag{2.2.11}$$

The following properties hold for $\mathcal{M}$ and $\mathcal{N}$.

Proposition 2.2.1. *Let $\phi, \psi : \mathbb{R}^d \times \mathbb{I} \to \mathbb{R}$, and $\mathcal{M}$, $\mathcal{N}$ be defined by* (2.2.10)-(2.2.11)*. Then:*

1. *If $\phi \le \psi$, then $\mathcal{M}\phi \le \mathcal{M}\psi$*
2. $\mathcal{M}(t\phi + (1-t)\psi) \ge t\mathcal{M}\phi + (1-t)\mathcal{M}\psi \quad \forall t \in [0,1]$
3. $\mathcal{M}(\phi + c) = \mathcal{M}\phi + c \quad \forall c \in \mathbb{R}$
4. $|\mathcal{M}\phi - \mathcal{M}\psi|_0 \le |\phi - \psi|_0$

The same properties also hold for the operator $\mathcal{N}$.

Remark 2.2.1. *Properties 1 to 3 are similar to the ones from [36] and are valid by definition of $\mathcal{M}$ and $\mathcal{N}$. Property 4 amounts saying that the operator $\mathcal{M}$ is nonexpansive, this is a consequence of assumptions (A4) and (A7).*

Now we go back to the characterization of the value function V. Our goal is to show that V solves

$$\begin{cases} \lambda V(x,q) + H\big(x,q,D_x V(x,q)\big) = 0 & \text{on } (\mathbb{R}^d \times \mathbb{I}) \setminus (\mathcal{A} \cup \mathcal{C}) \\ \max\big\{\lambda V(x,q) + H\big(x,q,D_x V(x,q)\big), \\ \qquad V(x,q) - \mathcal{N}V(x,q)\big\} = 0 & \text{on } \mathcal{C} \\ V(x,q) - \mathcal{M}V(x,q) = 0 & \text{on } \mathcal{A}. \end{cases} \tag{2.2.12}$$

Unfortunately this system doesn't admit a C^1 solution in general, and we need to define one in a weaker sense. For this purpose, we recall the definition of *viscosity solution* used in [9].

Definition 2.2.2 (Viscosity solution)**.** *Assume (A1)–(A7). Let $w : \mathbb{R}^d \times \mathbb{I} \to \mathbb{R}$ be a bounded and uniformly continuous function. We say that w is a viscosity sub-[respectively super-]solution of the HJB equation* (2.2.12) *if, for any bounded function $\phi : \mathbb{R}^d \to \mathbb{R}$ with continuous and bounded first derivative, the following property holds.*

For any $q \in \mathbb{I}$, at each local maximum [resp. minimum] point (x', q) of $w(x, q) - \phi(x)$ we have

$$\begin{cases} \lambda V(x',q) + H(x',q,D_x\phi(x')) \leq 0 \quad [\textit{resp.} \geq 0] & \textit{on } (\mathbb{R}^d \times \mathbb{I}) \setminus (\mathcal{A} \cup \mathcal{C}) \\ \max\Big\{\lambda V(x',q) + H(x',q,D_x\phi(x')), & \\ \qquad V(x',q) - \mathcal{N}V(x',q)\Big\} \leq 0 \quad [\textit{resp.} \geq 0] & \textit{on } \mathcal{C} \\ V(x',q) - \mathcal{M}V(x',q) \leq 0 \quad [\textit{resp.} \geq 0] & \textit{on } \mathcal{A} \end{cases}$$

A viscosity solution is a function which is simultaneously sub- and super-solution.

The previous definition allows us to characterize V.

Proposition 2.2.2. *Assume (A1)–(A7). The function V is a bounded and Hölder continuous viscosity solution of:*

$$\begin{cases} \lambda V(x,q) + H(x,q,D_xV(x,q)) = 0 & \textit{on } (\mathbb{R}^d \times \mathbb{I}) \setminus (\mathcal{A} \cup \mathcal{C}) \quad (2.2.13a) \\ \max\{\lambda V(x,q) + H(x,q,D_xV(x,q)), & \\ \qquad V(x,q) - \mathcal{N}V(x,q)\} = 0 & \textit{on } \mathcal{C} \quad (2.2.13b) \\ V(x,q) - \mathcal{M}V(x,q) = 0 & \textit{on } \mathcal{A}. \quad (2.2.13c) \end{cases}$$

The proof is given in [26, Theorem 4.2]. The same arguments of the proof of Theorem 5.1 in [26] can then be used to obtain a strong comparison principle (and hence, uniqueness of the solution) as follows:

Theorem 2.2.1. *Assume (A1)–(A7). Let w [respectively v] be a bounded usc [resp. lsc] function on $\mathbb{R}^d$. Assume that w is a subsolution [resp. v is a supersolution] of* (2.2.13) *in the following sense: for any $q \in \mathbb{I}$*

$$\begin{cases} \lambda V(x,q) + H(x,q,D_xV(x,q)) \leq 0 \quad [\textit{resp.} \geq 0] & \textit{on } (\mathbb{R}^d \times \mathbb{I}) \setminus (\mathcal{A} \cup \mathcal{C}) \\ \max\Big\{\lambda V(x,q) + H(x,q,D_xV(x,q)), & \\ \qquad V(x,q) - \mathcal{N}V(x,q)\Big\} \leq 0 \quad [\textit{resp.} \geq 0] & \textit{on } \mathcal{C} \\ V(x,q) - \mathcal{M}V(x,q) \leq 0 \quad [\textit{resp.} \geq 0] & \textit{on } \mathcal{A} \end{cases}$$

Then, $w \leq v$.

The viscosity framework turns out to be a convenient tool for the study of the theoretical properties of the value function and also for the analysis of the convergence of numerical schemes.

Chapter 3

Approximation of the Hamilton-Jacobi-Bellman equation

3.1 The numerical scheme

Here we further the study of the numerical approximation of the value function by a class of schemes obtained as adaptations of monotone schemes to the hybrid case. The main goal of this chapter is to establish error estimates between the value function and its numerical approximation.

Consider monotone approximation schemes of (2.2.13a)-(2.2.13c), of the following form:

$$\begin{cases} S(h,x,q,V_h(x,q),V_h) = 0 & \text{on } (\mathbb{R}^d \times \mathbb{I}) \setminus (\mathcal{A} \cup \mathcal{C}) \quad (3.1.1a) \\ \max\Big\{S(h,x,V_h(x,q),V_h), \\ \qquad V_h(x,q) - \mathcal{N}V_h(x,q)\Big\} = 0 & \text{on } \mathcal{C} \quad (3.1.1b) \\ V_h(x,q) - \mathcal{M}V_h(x,q) = 0 & \text{on } \mathcal{A} \quad (3.1.1c) \end{cases}$$

Here $S : \mathbb{R}^d_+ \times \mathbb{R}^d \times \mathbb{I} \times \mathbb{R} \times C_{\mathrm{b}}(\mathbb{R}^d \times \mathbb{I}) \to \mathbb{R}$ is a consistent, monotone operator which is considered to be an approximation of the HJB equation (2.2.13a) (see assumptions (S1)-(S3) for the precise properties). We will denote by $h \in \mathbb{R}^d_+$ the mesh size, and by $V_h \in C_{\mathrm{b}}(\mathbb{R}^d \times \mathbb{I})$ the solution of (3.1.1). The rest of the chapter is devoted to study conditions on S under which V_h is an approximation of V.

The abstract notations of the scheme have been introduced by Barles and Souganidis [6] to display the monotonicity of the scheme: $S(h,x,q,r,v)$ is non decreasing in r and non increasing in v. Typical approximation schemes that can be put in this framework are finite differences methods [40, 64], Semi-Lagrangian schemes [27, 16], and Markov chain approximations [40].

In all the sequel, we make the following assumptions on the discrete scheme (3.1.1):

(S1) Monotonicity: for all $h \in \mathbb{R}^d_+$, $m \geq 0$, $x \in \mathbb{R}^d$, $r \in \mathbb{R}$, $q \in \mathbb{I}$, and ϕ, ψ in $C_{\mathrm{b}}(\mathbb{R}^d)$ such that $\phi \leq \psi$ in $\mathbb{R}^d$

$$S(h, x, q, r, \phi + m) \leq m + S(h, x, q, r, \psi)$$

(S2) Regularity: for all $h \in \mathbb{R}^d_+$ and $\phi \in C_{\mathrm{b}}(\mathbb{R}^d)$, $x \mapsto S(h, x, q, r, \phi)$ is bounded and continuous. For any $R > 0$, $r \mapsto S(h, x, q, r, \phi)$ is uniformly continuous on the ball $B(0, R)$ centered at 0 and with radius R, uniformly with respect to $x \in \mathbb{R}^d$.

(S3) Consistency: There exist $n, k_i > 0$, $i \in J \subseteq \{1, \ldots, n\}$ and a constant $K_c > 0$ such that for all $h \in \mathbb{R}^d_+$ and x in $\mathbb{R}^d$, and for every smooth $\phi \in C^n(\mathbb{R}^d)$ such that $|D^i\phi|_0$ is bounded, for every $i \in J$ and $q \in \mathbb{I}$, the following holds:

$$\left|\lambda\phi(x) + H(x, q, D\phi(x)) - S(h, x, q, \phi(x), \phi)\right| \leq K_c \mathcal{E}(h, \phi)$$

where $\mathcal{E}(h, \phi) := \sum_{i \in J} |D^i\phi|_0 |h|^{k_i}$. Here $D^i\phi$ denotes the i-th derivative of the function ϕ.

We also assume that for each $h \in \mathbb{R}^d_+$, the numerical scheme has a unique solution V_h. Then, combining consistency, monotonicity and L^∞ stability, it is a standard matter to recover convergence by means of Barles-Souganidis theorem:

Theorem 3.1.1 (Barles and Souganidis [6])**.** *Assume (A1)-(A7) and let $V \in C_{b,l}(\mathbb{R}^d \times \mathbb{I})$ be the viscosity solution of* (2.2.13)*. Assume (S1)-(S3) and that* (3.1.1) *admits a unique solution $V_h \in C_b(\mathbb{R}^d \times \mathbb{I})$. Then V_h converges locally uniformly to V.*

We point out that the existence of such a numerical scheme holds for many classical monotone schemes. For example, the existence of V_h has been established in [28] for a class of Semi-Lagrangian (SL) schemes.

In the sequel, we define a more precise framework where the scheme (3.1.1) admits a unique solution.

3.2 Policy Iteration and Semi-Lagrangian schemes

In this section, we study efficient numerical methods for applying Dynamic Programming techniques to hybrid optimal control problems of infinite horizon type.

From the very start of Dynamic Programming techniques [8, 35], Policy Iteration (PI) has been recognized as a viable, usually faster alternative to

Value Iteration (VI) in computing the fixed point of the Bellman operator. Among the wide literature on PI, we quote here the pioneering theoretical analysis of Puterman and Brumelle [55], which have shown that the linearization procedure underlying PI is equivalent to a Newton-type iterative solver. More recently, the abstract setting of [55] has been adapted to computationally relevant cases [59], proving superlinear (and, in some cases, quadratic) convergence of PI. Moreover, we mention that an adaptation of PI to large sparse problems has been proposed as Modified Policy Iteration (MPI) in [56], and has also become a classical tool. It's important to remark that PI, however, is not the only approach for accelerating the solution process of numerical schemes: in the case of Finite Difference schemes, for instance, one might consider improving performances via Fast Methods [63, 19].

In the present chapter, we intend to study the construction and numerical validation of a SL scheme with PI/MPI sover for hybrid optimal control. To this end, we will recall the general algorithm, sketch some implementation details for the simple case of one-dimensional dynamics, and test the scheme on some numerical examples in dimension $d = 1, 2$.

The outline of the chapter is the following. In Section 3.3 we review the main results about the Bellman equation characterizing the value function, and construct a Semi-Lagrangian approximation for V in the form of value iteration. In Section 3.4 we improve the algorithm by using a policy iteration technique. Finally, Section 3.5 presents some numerical examples of approximation of the value function as well as the construction of the optimal control.

3.3 A Semi-Lagrangian scheme for hybrid control problems

First, we recall some basic analytical results about the value function (2.2.4). To this end, we start by making a precise set of assumptions on the problem.

3.3.1 Numerical approximation

In order to set up a numerical approximation for (2.2.13), we construct a discrete grid of nodes (x_i, q) in the state space and fix the discretization parameters Δx and Δt. In what follows, we denote the discretization steps in compact form by $h := (\Delta t, \Delta x)$ and the approximate value function by V_h.

Following [28], we write the fixed point form of the scheme at (x_i, q) as

$$v_i^{(q)} = V_h(x_i, q) = \begin{cases} M^h V_h(x_i, q) & (x_i, q) \in \mathcal{A} \\ \min\left\{\Sigma^h(x_i, q, V_h), N^h V_h(x_i, q)\right\} & (x_i, q) \in \mathcal{C} \\ \Sigma^h(x_i, q, V_h) & \text{else} \end{cases} \tag{3.3.1}$$

in which N^h, M^h and Σ^h are consistent and monotone numerical approximations for, respectively, the operators $\mathcal{N}$, $\mathcal{M}$ and the Hamiltonian H. More compactly, (3.3.1) could be written as

$$V_h = T^h(V_h).$$

We recall that, for $\lambda > 0$, under the basic assumption which ensure continuity of the value function, the right-hand side of (3.3.1) is a contraction [28] and can therefore be solved by fixed-point iteration, also known as *value iteration* (VI):

$$V_h^{(j+1)} = T^h(V_h^{(j)}). \tag{3.3.2}$$

In order to define the scheme more explicitly, as well as to extend the approximate value function to all $x \in \mathbb{R}^d$ and $q \in \mathbb{I}$, we use a monotone interpolation $\mathcal{I}$ constructed on the node values, and denote by $\mathcal{I}[V_h](x,q)$ the interpolated value of V_h computed at (x,q).

We recall the definition of a monotone interpolation, adapted to our case.

Definition 3.3.1 (Monotone interpolation)**.** *Let $\mathcal{I}$ be an interpolation operator. $\mathcal{I}$ is said to be monotone if, given two functions $\phi, \psi : \mathbb{R}^d \times \mathbb{I} \to \mathbb{R}$ such that $\phi(x,q) \geq \psi(x,q)$ for every $(x,q) \in \mathbb{R}^d \times \mathbb{I}$ the following holds*

$$\mathcal{I}[\phi](x,q) \geq \mathcal{I}[\psi](x,q) \quad \forall (x,q) \in \mathbb{R}^d \times \mathbb{I}$$

With this notation, a natural definition of the discrete jump operators M^h and N^h is given by

$$M^h V_h(x,q) := \min_{\nu \in \mathcal{V}} \Big\{ \mathcal{I}[V_h](g(x,q,\nu)) + c_{\mathcal{A}}(x,q,\nu) \Big\} \tag{3.3.3}$$

$$N^h V_h(x,q) := \min_{(x',q') \in \mathcal{D}} \Big\{ \mathcal{I}[V_h](x',q') + c_{\mathcal{C}}(x,q,x',q') \Big\}. \tag{3.3.4}$$

A standard Semi-Lagrangian discretization of the Hamiltonian related to continuous control is provided (see [27]) by

$$\Sigma^h(x,q,V_h) := \min_{u \in U} \Big\{ \Delta t \ell(x,q,u) + e^{-\lambda \Delta t} \mathcal{I}[V_h](x + \Delta t f(x_i,q,u), q) \Big\}. \tag{3.3.5}$$

In the SL form, the value iteration (3.3.2) might then be recast at a node (x_i,q) as

$$V_h^{(j+1)}(x_i,q) = \begin{cases} M^h V_h^{(j)}(x_i,q) & (x_i,q) \in \mathcal{A} \\ \min \Big\{ N^h V_h^{(j)}(x_i,q), \Sigma^h(x_i,q,V_h^{(j)}) \Big\} & (x_i,q) \in \mathcal{C} \\ \Sigma^h(x_i,q,V_h^{(j)}) & \text{else} \end{cases} \tag{3.3.6}$$

with Σ^h given by (3.3.5), and j denoting the iteration number.

We can also check that such a scheme satisfies all the required properties. Monotonicity, regularity and consistency of the jump operators can be obtained by using a monotone interpolation in the definition of the operators M^h and N^h. For example, $\mathbb{P}_1$ (piecewise linear) or $\mathbb{Q}_1$ (piecewise multilinear) interpolations would satisfy the requirements in Definition 3.3.1, but the choice is not limited to those two.

Regarding the operator Σ^h, its monotonicity and regularity also come from the choice of a monotone interpolation of the value function V_h. As for its consistency, we start by giving the following estimate on the interpolation term in (3.3.5). In the case of a $\mathbb{P}_1$ interpolation, the approximation error is $O(\Delta x^2)$. Hence, for every smooth function ϕ satisfying the properties described in (S3), we have, for some constants $C_1 > 0$ and $C_2 > 0$

$$\begin{aligned}
&\mathcal{I}[\phi](x + \Delta t f(x,q,u), q) = \\
&\quad = \phi(x + \Delta t f(x,q,u), q) + C_1 |D^2\phi|_0 \Delta x^2 = \\
&\quad = \phi(x,q) + \Delta t f(x,q,u) \cdot D\phi(x,q) + C_2 |D^2\phi|_0 (\Delta t^2 + \Delta x^2).
\end{aligned} \tag{3.3.7}$$

Our scheme takes the form

$$S(\Delta t, x, q, \phi(x), \phi) := \frac{\phi(x,q) - \Sigma(x,q,\phi)}{\Delta t}$$

and, by applying (3.3.7) and $e^{-\lambda \Delta t} = 1 - \lambda \Delta t + O(\Delta t^2)$, we obtain

$$\begin{aligned}
&\frac{\phi(x,q) - \Sigma(x,q,\phi)}{\Delta t} = \\
&= \frac{1}{\Delta t} \sup_{u \in U} \left\{ \phi(x,q) - \Delta t \ell(x,q,u) - e^{-\lambda \Delta t} \mathcal{I}[\phi](x + \Delta t f(x,q,u), q) \right\} = \\
&= \frac{1}{\Delta t} \sup_{u \in U} \Big\{ \phi(x,q) - \Delta t \ell(x,q,u) - \\
&\quad - \Big(\phi(x,q) + \Delta t f(x,q,u) \cdot D\phi(x,q) + C_2 |D^2\phi|_0 (\Delta t^2 + \Delta x^2) \Big) + \\
&\quad + \lambda \Delta t \Big(\phi(x,q) + \Delta t f(x,q,u) \cdot D\phi(x,q) + C_2 |D^2\phi|_0 (\Delta t^2 + \Delta x^2) \Big) + \\
&\quad + O(\Delta t^2) \Big\} = \\
&= \sup_{u \in U} \{ \lambda \phi(x,q) - \ell(x,q,u) - f(x,q,u) \cdot D\phi(x,q) \} + \\
&\quad + C_3 |D\phi|_0 \Delta t + C_2 |D^2\phi|_0 \Delta t + O\left(\frac{\Delta x^2}{\Delta t} \right).
\end{aligned} \tag{3.3.8}$$

for some constants and $C_3 > 0$. Now, under the additional assumption $\frac{\Delta x^2}{\Delta t} \to 0$ (also known as inverse CFL condition) by the definition of the Hamiltonian H and from (3.3.8), we can recover the consistency property

(S3):

$$S(\Delta t, x, q, \phi(x), \phi) = \lambda\phi(x,q) + H\big(x, q, D\phi(x,q)\big) + K_{\mathrm{c}} \sum_{i\in\{1,2\}} |D^i\phi|_0 \Delta t$$

for some constant $K_{\mathrm{c}} > 0$ independent of Δt or Δx.

3.4 Policy Iteration algorithm

Following [60], we give now an even more explicit form of the scheme, which is the one applied to the one-dimensional examples of Section 3.5. Once we set up a one-dimensional space grid of evenly spaced nodes $x_1, \dots, x_n$ with space step Δx, the discrete solution may be given by the column vector

$$\boldsymbol{v} := (\boldsymbol{v}^{(1)}, \boldsymbol{v}^{(2)}, \dots, \boldsymbol{v}^{(m)})^{\intercal} \in \mathbb{R}^{nm}.$$

This vector is constructed by concatenating the column vectors $\boldsymbol{v}^{(q)} := (v_1^{(q)}, \dots, v_n^{(q)})$, each denoting the discretized value function associated to the q-th component of the state space. Within the vector $\boldsymbol{v}$, the element $v_i^{(k)}$ appears with the index $(k-1)n + i$.

Keeping the same notation for all vectors, $\boldsymbol{u} \in U^{nm}$ will denote the vector of controls of the system, $u_i^{(k)}$ being the value of the control at the space node x_i while the k-th dynamics is active. We also define the vector $\boldsymbol{s} \in \mathbb{I}^{nm}$ representing the switching strategy, so that $s_i^{(k)} = l$ means that if the trajectory is in x_i and the active dynamics is k, the system commutes from k to l. Note that, in the numerical examples of Sec. 3.5, discontinuous jumps will always appear only on the discrete component of the state space. For example, we have $x' = x$ and this data need not be kept in memory (we will use the term *switch* to denote a state transition of this kind).

In the general case, we would also need to keep memory of the arrival point of the jump and/or of the discrete control w in the case of an autonomous jump. In general, the arrival point is not a grid point, so that we also need to perform an interpolation in (3.3.3)-(3.3.4). Therefore, the details for the general case can be recovered by mixing the basic arguments used in what follows.

The endpoint of this construction is to put the problem in the standard form used in Policy Iteration,

$$\min_{(\boldsymbol{u},\boldsymbol{s})\in U^{nm}\times\mathbb{I}^{nm}} \big\{B(\boldsymbol{u},\boldsymbol{s})\boldsymbol{v} - \boldsymbol{c}(\boldsymbol{u},\boldsymbol{s})\big\} = 0, \tag{3.4.1}$$

with explicitly defined matrix B and vector $\boldsymbol{c}$. Note that, in (3.4.1), we have made clear the fact that a policy is composed of both a feedback control $\boldsymbol{u}$ and a switching strategy $\boldsymbol{s}$.

For every given $\boldsymbol{s}$, define now the matrices $D_{\mathcal{A}}(\boldsymbol{s}), D_{\mathcal{C}}(\boldsymbol{s}) \in M_{nm}(\{0,1\})$ as permutations of the array $\boldsymbol{v}$. These matrices represent changes in the state

due to the switching strategy: $D_{\mathcal{A}}(\boldsymbol{s})$ corresponds to autonomous jumps and $D_{\mathcal{C}}(\boldsymbol{s})$ to controlled jumps. Note that, in our case, the elements of $D_{\mathcal{A}}(\boldsymbol{s})$ and $D_{\mathcal{C}}(\boldsymbol{s})$ will be in $\{0,1\}$, that there exists at most one nonzero element on each row, and that the two matrices cannot have a nonzero element in the same position.

In order to determine the positions of the nonzero elements in the matrix $D_{\mathcal{A}}(\boldsymbol{s})$, we apply the following rule. For all $(i,k) \in \{1,\dots,n\} \times \mathbb{I}$, if the following conditions hold:

$$\begin{cases} (x_i,k) \in \mathcal{A} & \\ s_i^{(k)} \neq k & \text{(a switch occurs)} \\ (x_i, s_i^{(k)}) \in g(x_i,k,\mathcal{V}) & \text{(the switch is in the image of } g), \end{cases}$$

then, defining the indices a and b

$$\begin{cases} a = (k-1)n + i \\ b = (s_i^{(k)} - 1)n + i \end{cases}$$

the element in row a and column b of the matrix $D_{\mathcal{A}}(\boldsymbol{s})$, denoted $[D_{\mathcal{A}}]_{a,b}(\boldsymbol{s})$, is equal to 1.

Similarly, the nonzero elements of the matrix $D_{\mathcal{C}}(\boldsymbol{s})$ are defined by the following rule. For all $(i,k) \in \{1,\dots,n\} \times \mathbb{I}$, if the following conditions hold:

$$\begin{cases} (x_i,k) \in \mathcal{C} & \\ s_i^{(k)} \neq k & \text{(a switch occurs)}, \end{cases}$$

then, defining the indices a and b

$$\begin{cases} a = (k-1)n + i \\ b = (s_i^{(k)} - 1)n + i \end{cases}$$

the element in row a and column b of the matrix $D_{\mathcal{C}}(\boldsymbol{s})$, denoted $[D_{\mathcal{C}}]_{a,b}(\boldsymbol{s})$, is equal to 1.

Last, we define the matrix

$$D(\boldsymbol{s}) := D_{\mathcal{A}}(\boldsymbol{s}) + D_{\mathcal{C}}(\boldsymbol{s})$$

which accounts for changes in the state related to the switching strategy, both autonomous and controlled.

We turn now to the continuous control part. First, we write Σ in vector form as

$$\boldsymbol{\Sigma}(\boldsymbol{x},k,\boldsymbol{v}) = \min_{\boldsymbol{u}^{(k)} \in U^n} \left\{ \Delta t\, \boldsymbol{\ell}(\boldsymbol{x},k,\boldsymbol{u}^{(k)}) + e^{-\lambda \Delta t} \mathcal{E}(\boldsymbol{x},k,\boldsymbol{u}^{(k)}) \boldsymbol{v}^{(k)} \right\}$$

where the matrix $\mathcal{E}(\boldsymbol{x},k,\boldsymbol{u}^{(k)}) \in M_n(\mathbb{R})$ is defined so as to have

$$\mathcal{E}(\boldsymbol{x},k,\boldsymbol{u}^{(k)}) \boldsymbol{v}^{(k)} = \boldsymbol{\mathcal{I}}[V_h](\boldsymbol{x} + \Delta t\, f(\boldsymbol{x},k,\boldsymbol{u}^{(k)}),k)$$

and $\boldsymbol{\mathcal{I}}[V_h](\boldsymbol{x}+\Delta t\, f(\boldsymbol{x},k,\boldsymbol{u}^{(k)}),k)$ and $\boldsymbol{\ell}(\boldsymbol{x},k,\boldsymbol{u}^{(k)})$ denote vectors which collect respectively all the values $\mathcal{I}[V_h](x_i+\Delta t f(x_i,k,u_i^{(k)}),k)$ and $\ell(x_i,k,u_i^{(k)})$.

At internal points, using a monotone $\mathbb{P}_1$ interpolation for the values of $\boldsymbol{v}$ results in a convex combination of node values. On the boundary of the domain, the well-posedness of the problem requires either to have an invariance condition (which implies that $f(x_i,k,u_i^{(k)})$ always points inwards) or to perform an autonomous jump or switch when the boundary is reached. Therefore, we should not care about defining a space reconstruction outside of the computational domain, although this could be accomplished by extrapolating the internal values.

For every $\boldsymbol{u}$ and $\boldsymbol{s}$, the matrix $E(\boldsymbol{u},\boldsymbol{s})\in M_{nm}(\mathbb{R})$ is then constructed in the block diagonal form:

$$E(\boldsymbol{u},\boldsymbol{s}) := \begin{pmatrix} E^{(1)}(\boldsymbol{u}^{(1)},\boldsymbol{s}^{(1)}) & 0 & \cdots & 0 \\ 0 & E^{(2)}(\boldsymbol{u}^{(2)},\boldsymbol{s}^{(2)}) & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & E^{(m)}(\boldsymbol{u}^{(m)},\boldsymbol{s}^{(m)}) \end{pmatrix}.$$

Assuming for simplicity that we work at Courant numbers below the unity (although this is not necessary for the stability of SL schemes), each block $E^{(k)}(\boldsymbol{u}^{(k)},\boldsymbol{s}^{(k)})\in M_n(\mathbb{R})$ is a sparse matrix with non-zero elements $E_{i,j}^{(k)}$ determined so as to implement a $\mathbb{P}_1$ space interpolation, in the following way: for every $(i,k)\in\{1,\dots,n\}\times\mathbb{I}$, define

$$h_{i,k} := \frac{\Delta t}{\Delta x} f(x_i,k,u_i^{(k)})$$

and

$$\text{if}\quad \begin{cases} s_i^{(k)}=k \\ h_{i,k}<0 \end{cases} \qquad \text{then}\quad \begin{cases} E_{i,i-1}^{(k)}(u_i^{(k)},s_i^{(k)}) = 1+h_{i,k} \\ E_{i,i}^{(k)}(u_i^{(k)},s_i^{(k)}) = -h_{i,k} \end{cases}$$

else,

$$\text{if}\quad \begin{cases} s_i^{(k)}=k \\ h_{i,k}>0 \end{cases} \qquad \text{then}\quad \begin{cases} E_{i,i}^{(k)}(u_i^{(k)},s_i^{(k)}) = 1-h_{i,k} \\ E_{i,i+1}^{(k)}(u_i^{(k)},s_i^{(k)}) = h_{i,k}. \end{cases}$$

Note that, if a switching strategy $\boldsymbol{z}\in\mathbb{I}^{nm}$ doesn't perform any switch (i.e. $z_i^{(k)}=k$ for all $(i,k)\in N\times\mathbb{I}$), by definition of the matrix $E(\boldsymbol{u},\boldsymbol{s})$ we obtain, for all $k\in\mathbb{I}$,

$$E^{(k)}(\boldsymbol{u}^{(k)},\boldsymbol{z}^{(k)})\boldsymbol{v}^{(k)} = \mathcal{E}(\boldsymbol{x},k,\boldsymbol{u}^{(k)})\boldsymbol{v}^{(k)}$$

whereas, in the general case, if a switch occurs at x_i, then the corresponding element of the matrix $E^{(k)}$ is zero. Finally, we define the vector $\boldsymbol{c}(\boldsymbol{u},\boldsymbol{s})\in\mathbb{R}^{nm}$ with a block structure of the form:

$$\boldsymbol{c}(\boldsymbol{u},\boldsymbol{s}) = (\boldsymbol{c}^{(1)}(\boldsymbol{u}^{(1)},\boldsymbol{s}^{(1)}),\boldsymbol{c}^{(2)}(\boldsymbol{u}^{(2)},\boldsymbol{s}^{(2)}),\dots,\boldsymbol{c}^{(m)}(\boldsymbol{u}^{(m)},\boldsymbol{s}^{(m)}))$$

with $\boldsymbol{c}^{(k)}(\boldsymbol{u}^{(k)}, \boldsymbol{s}^{(k)}) \in \mathbb{R}^n$ such that, for every (i,k) in $\{1,\dots,n\} \times \mathbb{I}$,

$$c_i^{(k)}(u_i^{(k)}, s_i^{(k)}) = \begin{cases} -\Delta t\, \ell(x_i, k, u_i^{(k)}) & s_i^{(k)} = k \\ -\xi(k, s_i^{(k)}) & s_i^{(k)} \neq k \end{cases} \tag{3.4.2}$$

where $\xi(k,l)$ denotes the switching cost ($c_{\mathcal{A}}$ or $c_{\mathcal{C}}$) from dynamics k to l.

With these notations, we can write the SL scheme (3.3.1) in vector form as

$$\boldsymbol{v} = \min_{(\boldsymbol{u},\boldsymbol{s}) \in U^{nm} \times \mathbb{I}^{nm}} \left\{ [D(\boldsymbol{s}) + e^{-\lambda \Delta t} E(\boldsymbol{u}, \boldsymbol{s})]\boldsymbol{v} - \boldsymbol{c}(\boldsymbol{u}, \boldsymbol{s}) \right\} \tag{3.4.3}$$

or, defined the matrix

$$B(\boldsymbol{u}, \boldsymbol{s}) := -I_{nm} + D(\boldsymbol{s}) + e^{-\lambda \Delta t} E(\boldsymbol{u}, \boldsymbol{s})$$

as

$$\min_{(\boldsymbol{u},\boldsymbol{s}) \in U^{nm} \times \mathbb{I}^{nm}} \{ B(\boldsymbol{u}, \boldsymbol{s})\boldsymbol{v} - \boldsymbol{c}(\boldsymbol{u}, \boldsymbol{s}) \} = 0.$$

Once we have reformulated the Semi-Lagrangian scheme for the hybrid control problem in the standard form, we can solve it using Algorithm 1. The only difference with a standard PI algorithm is the inclusion of the switching strategy in the control policy.

Algorithm 1 Policy Iteration

$j \leftarrow 0$
STOP $\leftarrow$ FALSE
$\boldsymbol{u}_0 \in U^{nm}$
$\boldsymbol{s}_0 \in \mathbb{I}^{nm}$
while STOP = FALSE **do**
  if [stopping criterion satisfied] **then**
    STOP $\leftarrow$ TRUE
  else
    $\boldsymbol{v}_j \leftarrow \boldsymbol{w}$ solution of $B(\boldsymbol{u}_j, \boldsymbol{s}_j)\boldsymbol{w} = \boldsymbol{c}(\boldsymbol{u}_j, \boldsymbol{s}_j)$ (*Policy Evaluation*)
    $(\boldsymbol{u}_{j+1}, \boldsymbol{s}_{j+1}) \leftarrow \underset{(\boldsymbol{\mu},\boldsymbol{\sigma}) \in U^{nm} \times \mathbb{I}^{nm}}{\arg\min} \{ B(\boldsymbol{\mu}, \boldsymbol{\sigma})\boldsymbol{v}_j - \boldsymbol{c}(\boldsymbol{\mu}, \boldsymbol{\sigma}) \}$ (*Policy Improvement*)
    $j \leftarrow j + 1$
  end if
end while

The initialization of the control $\boldsymbol{u}_0$ and switching strategy $\boldsymbol{s}_0$ in Algorithm 1 (as well as its modified version, described later in Algorithm 2) is crucial to performances. A traditional approach consists in obtaining a first rough approximation of the value function $\boldsymbol{v}$ by performing a predetermined number of Value Iteration steps and then computing its minimum to have $\boldsymbol{u}_0$ and $\boldsymbol{s}_0$.

It is also important to detail the Policy Improvement step of the algorithm: in the presence of multiple minima, we have to make sure to implement a selection criterion.

We remark that some theoretical result obtained in the "classical" setting is also true in the hybrid setting. In particular, convergence might still be obtained by means of the theorem proven in [11]. In order to apply the result, we have to rewrite (3.4.1) according to the framework of [11], since in this paper problem (3.4.1) actually comes from a maximization problem. This can be done by simply changing the sign of B and $\boldsymbol{v}$ so that the new vector $\bar{\boldsymbol{v}} := -\boldsymbol{v}$ is the solution to

$$\min_{(\boldsymbol{\alpha},\boldsymbol{s})\in U^{nm}\times\mathbb{I}^{nm}} \left\{\bar{B}(\boldsymbol{\alpha},\boldsymbol{s})\bar{\boldsymbol{v}} - \boldsymbol{c}(\boldsymbol{\alpha},\boldsymbol{s})\right\} = 0. \tag{3.4.4}$$

where $\bar{B} := -B$.

With this notation, we can prove that the matrix $\bar{B}$ and the vector $\boldsymbol{c}$ satisfy the following properties:

(P1) For every $(\boldsymbol{u},\boldsymbol{s}) \in U^{nm}\times\mathbb{I}^{nm}$, the matrix $\bar{B}(\boldsymbol{u},\boldsymbol{s})$ is monotone. That is, if $\bar{B}(\boldsymbol{u},\boldsymbol{s})$ is invertible and $\bar{B}^{-1}(\boldsymbol{u},\boldsymbol{s}) \geq 0$ component wise.

(P2) The functions $\bar{B} : U^{nm}\times\mathbb{I}^{nm} \to M_{nm}(\mathbb{R})$ and $\boldsymbol{c} : U^{nm}\times\mathbb{I}^{nm} \to \mathbb{R}^{nm}$ are continuous.

The matrix $\bar{B} = -B$ is monotone for every $\boldsymbol{\alpha}$ ed $\boldsymbol{s}$ if and only if the system

$$\bar{B}(\boldsymbol{\alpha},\boldsymbol{s})\bar{\boldsymbol{w}} = \boldsymbol{b} \tag{3.4.5}$$

has a non-negative solution $\bar{\boldsymbol{w}}$ for every non-negative vector $\boldsymbol{b}$. Which is equivalent to proving that for every non-negative vector $\boldsymbol{b}$ the system

$$B(\boldsymbol{\alpha},\boldsymbol{s})\boldsymbol{w} = \boldsymbol{b} \tag{3.4.6}$$

has a non-positive solution $\boldsymbol{w}$. On the other hand, recasting (3.4.6) in fixed point form as

$$\boldsymbol{w} = \left[D(\boldsymbol{s}) + e^{-\lambda\Delta t}E(\boldsymbol{\alpha},\boldsymbol{s})\right]\boldsymbol{w} - \boldsymbol{b}. \tag{3.4.7}$$

we have (according to [28]) that, in the case of a monotone interpolation, the right hand side is a contraction and the solution can be recovered as the limit of the sequence

$$\boldsymbol{w}_{j+1} = \left[D(\boldsymbol{s}) + e^{-\lambda\Delta t}E(\boldsymbol{\alpha},\boldsymbol{s})\right]\boldsymbol{w}_j - \boldsymbol{b}.$$

Due to the monotonicity of the interpolation, the matrices D and E have non-negative elements and therefore preserve the sign of $\boldsymbol{w}_j$, whereas the vector $-\boldsymbol{b}$ is non-positive. Therefore, starting from an the initial vector $\boldsymbol{w}_0 = 0$, every element $\boldsymbol{w}_j$ of the sequence (and thus its limit $\boldsymbol{w}$) is non-positive. Hence, the matrix $\bar{B}$ is monotone.

The continuity of $\bar{B}$ and $\boldsymbol{c}$ is trivial if we provide the product space $U^{nm} \times \mathbb{I}^{nm}$ with the usual topology for $\boldsymbol{\alpha}$ with the discrete one for $\boldsymbol{s}$: a sequence $(\boldsymbol{\alpha}_j, \boldsymbol{s}_j)$ converges to $(\boldsymbol{\alpha}, \boldsymbol{s})$ if $\boldsymbol{\alpha}_j \to \boldsymbol{\alpha}$ and $\boldsymbol{s}_j$ is definitely equal to $\boldsymbol{s}$.

Now that we know that (P1) and (P2) are satisfied, we are ready to state the convergence theorem.

Theorem 3.4.1 ([11])**.** *Assume (P1)-(P2) hold. Then there exist a unique* $\bar{\boldsymbol{v}} \in \mathbb{R}^{nm}$ *solution of* (3.4.4)*. Moreover, the sequence* $\{\bar{\boldsymbol{v}}_j\}$ *generated by Algorithm 1 satisfies the following:*

i) $\bar{\boldsymbol{v}}_j \leq \bar{\boldsymbol{v}}_{j+1}$ *for every* $j \geq 0$*;*

ii) $\bar{\boldsymbol{v}}_j \to \bar{\boldsymbol{v}}$ *as* j *tends to* $+\infty$*.*

Under the same hypotheses, we can apply another result proven in [11] to obtain superlinear convergence.

Theorem 3.4.2 ([11])**.** *Assume (P1)-(P2) hold. Then problem* (3.4.4) *has a unique* $\bar{\boldsymbol{v}} \in \mathbb{R}^{nm}$ *and, for every initial guess* $(\boldsymbol{u}_0, \boldsymbol{s}_0) \in U^{nm} \times \mathbb{I}^{nm}$*, Algorithm 1 converges globally, i.e.*

$$\lim_{j \to +\infty} |\bar{\boldsymbol{v}}_j - \bar{\boldsymbol{v}}| = 0$$

and superlinearly, i.e.

$$|\bar{\boldsymbol{v}}_{j+1} - \bar{\boldsymbol{v}}| = o(|\bar{\boldsymbol{v}}_j - \bar{\boldsymbol{v}}|) \quad as \quad k \to +\infty$$

3.4.1 Modified policy iteration

A different iterative solver for the numerical scheme has been first proposed and analyzed in [56], and it is known as Modified Policy Iteration. It consists in performing the minimization in (3.3.6) only once every N_{it} iterations. In other terms, the policy evaluation step is replaced by N_{it} iterations of linear advection (in which, however, the transport may occur among different components of the state space). For $N_{\text{it}} = 1$ we obtain the value iteration, whereas for $N_{\text{it}} \to \infty$ the transport steps converge to an exact policy evaluation, and the algorithm coincides with the previous "exact" PI algorithm.

The pseudo-code in Algorithm 2 shows the MPI algorithm, for a comparison with the exact algorithm (Algorithm 1).

Note that, in the numerical test section, the MPI algorithm has been applied to the two-dimensional examples. Although the formulation in dimension $d = 2$ could be accomplished by a suitable redefinition of the vectors and matrices, in practice the MPI algorithm does not need such a formalism.

Concerning convergence, the hybrid case can again be treated with the same theoretical tools of the original proof in [56], which relies on the monotonicity of the (discretized) Bellman operator, as well as on giving an upper

Algorithm 2 Modified Policy Iteration

$j \leftarrow 0$
STOP $\leftarrow$ FALSE
$\boldsymbol{u}_0 \in U^{nm}$
$\boldsymbol{s}_0 \in \mathbb{I}^{nm}$
while STOP = FALSE **do**
  if [stopping criterion satisfied] **then**
    STOP $\leftarrow$ TRUE
  else
    if $j = 0 \pmod{N_{\text{it}}}$ **then**
      $(\boldsymbol{u}_{j+1}, \boldsymbol{s}_{j+1}) \leftarrow \underset{(\boldsymbol{\mu},\boldsymbol{\sigma})\in U^{nm}\times\mathbb{I}^{nm}}{\arg\min} \{B(\boldsymbol{\mu},\boldsymbol{\sigma})\boldsymbol{v}_j - \boldsymbol{c}(\boldsymbol{\mu},\boldsymbol{\sigma})\}$
      (*Policy Improvement*)
    else
      $(\boldsymbol{u}_{j+1}, \boldsymbol{s}_{j+1}) \leftarrow (\boldsymbol{u}_j, \boldsymbol{s}_j)$
    end if
    $\boldsymbol{v}_{j+1} \leftarrow \left[D(\boldsymbol{s}_{j+1}) + e^{-\lambda\Delta t}E(\boldsymbol{u}_{j+1}, \boldsymbol{s}_{j+1})\right]\boldsymbol{v}_j - \boldsymbol{c}(\boldsymbol{u}_{j+1}, \boldsymbol{s}_{j+1})$
    (*Inexact Policy Evaluation*)
    $j \leftarrow j + 1$
  end if
end while

and a lower bound on the sequence $\boldsymbol{v}_j$ by means of two converging sequences (one of which generated by value iteration). More precisely, the sequence considered in the convergence proof for the MPI is the sequence of approximations obtained after each policy improvement. In our notation, this is the subsequence $\boldsymbol{v}_l$ corresponding to $j = lN_{\text{it}} + 1$. We have therefore the following.

Theorem 3.4.3 ([56]). *Let $\boldsymbol{v}$ be the solution of* (3.4.3)*, and $\boldsymbol{v}_j$ be defined by Algorithm 2. If*

$$\min_{\boldsymbol{u},\boldsymbol{s}} \{B(\boldsymbol{u},\boldsymbol{s})\boldsymbol{v}_0 - \boldsymbol{c}(\boldsymbol{u},\boldsymbol{s})\} \leq 0$$

then, for any $N_{it} \geq 1$, the subsequence $\boldsymbol{v}_l$ obtained for $j = lN_{it} + 1$ is monotone decreasing, and $\boldsymbol{v}_l \to \boldsymbol{v}$ for $l \to \infty$.

3.5 Numerical tests

We give in this section some numerical examples in one and two space dimensions, comparing the performances of Value and Policy Iteration (exact PI algorithm in one dimension, and MPI in two dimensions). The comparison shows a substantial improvement in the convergence of the solver for the exact PI algorithm, whereas the MPI performs roughly the same number of

iterations as the VI. Here, the bottleneck is apparently the contraction coefficient of the Bellman operator. Nevertheless, the MPI allows to avoid the minimization step in a large majority of the iterates, thus reducing the CPU time. Note that in both two-dimensional examples the control appears only as a switching strategy, and the complexity of policy evaluation steps is reduced by a factor $1/m$. For more complex control actions the improvement in computing time would be even greater.

We also remark that the following tests do not exceed dimension 2 because their purpose is to show practical applications of the PI/MPI algorithm to hybrid control problems. Tackling the problem of the complexity arising from the increase of the problem's dimension is beyond the scope of this thesis, since the gain in performance obtainable with these algorithms is not sufficient to overcome the *Curse of Dimensionality*.

3.5.1 Stabilization of an unstable system

We now apply this technique to a stabilization problem: we consider a system with two dynamics: one "strong and expensive" and the other "weak and cheap". Only the former is able to keep the state of the system within the given set over time.

The state equation $\dot{X}(t) = f(X(t), Q(t), u(t))$ is defined by

$$f(x,q,u) = \begin{cases} x + d_1 u & q = 1 \\ x + d_2 u & q = 2 \end{cases}$$

where $d_1 < d_2$ and $-1 \leq u(t) \leq 1$ for every t in $[0, +\infty)$. Switching is mandatory only when the dynamics $q = 1$ is active and $|X(t)| = 1$, which implies that the state of the system belongs to the interval $[-1, 1]$ for all t in $[0, +\infty)$.

Here and in what follows, $c_{i,j}$ denotes a constant switching cost from $q = i$ to $q = j$, and the cost functional is defined as

$$\ell(x,q,u) = \begin{cases} x^2 + c_1 u^2 & q = 1 \\ x^2 + c_2 u^2 & q = 2. \end{cases}$$

The values assigned to all the parameters are summed up in Table 3.5.1, whereas Table 3.5.2 reports the number of iterations required for given stopping tolerances. In the first three examples, the stopping criterion reads

$$|\boldsymbol{v}_j - \boldsymbol{v}_{j-1}|_0 < \epsilon.$$

Note that, according to Table 3.5.2, squaring the tolerance makes the number of iteration N_P of the PI algorithm increase linearly, which indicates roughly quadratic convergence, while the number N_V for VI has a geometric behaviour as expected.

d_1	d_2	$c_{1,2}$	$c_{2,1}$	c_1	c_2	λ	t_f
0.5	2	0.2	0	0.25	4	1	20

Table 3.5.1: Choice of parameters, weak-strong test

ϵ	N_V	N_P
10^{-3}	456	8
10^{-6}	1147	10
10^{-12}	2786	12

Table 3.5.2: Number of iterations (VI and PI) for a given tolerance ϵ, weak-strong test

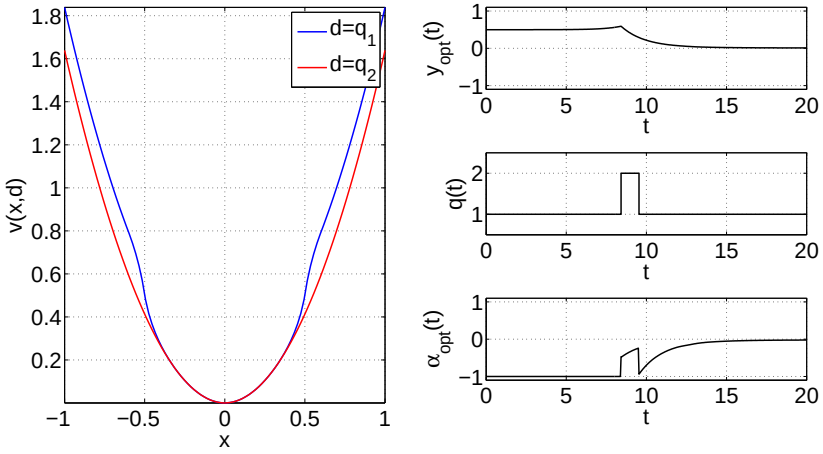


Figure 3.5.1: Value function, trajectory and optimal control, weak-strong test

Figure 3.5.1 shows the optimal strategy obtained for $\Delta t = 0.0067$, $\Delta x = \Delta t||f||_\infty$, $(X(0), Q(0)) = (0.5, 1)$, and $t \in [0, t_f]$. This strategy consists in using the unstable dynamics $q = 1$ as long as the state belongs to the interval $[-\bar{x}, \bar{x}]$ (where the value of $\bar{x} \in [-1, 1]$ depends on the given data). On the other hand, as soon as $|X(t)| > \bar{x}$, the optimal choice is to switch from $q = 1$ to $q = 2$ in order to stabilize the system and force it back towards the origin, then switch again to the first dynamics which can be used at a lower cost.

3.5.2 Three-gear vehicle

In this test, we consider the optimal control of a vehicle equipped with a three-gear engine, focusing on the acceleration strategy and the commutation between gears. Physical parameters correspond to the italian scooter Piaggio Vespa 50 Special.

m	ρ_1	ρ_2	ρ_3	r	c_d	τ	ν
140 [kg]	0.06	0.09	0.12	0.2 [m]	0.3	10 [Nm]	$6 \cdot 10^3$ [min^{-1}]

c_u	c_x	$c_{i,j}$	λ	t_f
1	0.5	$\begin{cases} 0.1 & i \neq j \\ 0 & i = j \end{cases}$	1	10 [s]

Table 3.5.3: Choice of parameters, three-gear vehicle test

ϵ	N_V	N_P
10^{-3}	337	6
10^{-6}	586	6
10^{-12}	1084	6

Table 3.5.4: Number of iterations (VI and PI) for a given tolerance ϵ, three-gear vehicle test.

The state equation for the speed of the vehicle is defined, for each gear $q \in \{1, 2, 3\}$, by

$$f(x, q, u) := \frac{1}{m}\left(\frac{T(\beta_q x)}{r\rho_q}u - c_{\mathrm{d}}x^2\right)$$

where $T(\omega) := \tau\left(\frac{\omega}{\nu} - \left(\frac{\omega}{\nu}\right)^3\right)$ is the power band of the engine, τ and ν are respectively its maximum torque and r.p.m., $\beta_q := \frac{60}{r\pi\rho_q}$ is a conversion coefficient with

$$\rho_q := \frac{\text{transmission shaft r.p.m.}}{\text{crankshaft r.p.m.}},$$

r is the radius of the wheel and c_d the drag coefficient. The control $u(t) \in [0, 1]$ represents the fraction of maximum torque used and the running cost is a linear combination of x and u:

$$\ell(x, u) = -c_x x + c_u u$$

c_x and c_u are positive weights. Lastly, we define $c_{i,j}$ as the switching cost from from $q = i$ to $q = j$.

The numerical results are obtained by assigning realistic values (Table 3.5.3) to the parameters defining the dynamics. The number of iterations is shown in Table 3.5.4 for various tolerances. The constant number of iterations obtained by PI might be due to the fact that optimal solutions (seem to) work with increasing values of q, this possibly meaning some sort of "causality" in the propagation of the value function.

Figure 3.5.2 shows the power band corresponding to our choice of τ and ν. Figure 3.5.3 shows the optimal solution obtained with $\Delta t = 0.027$ [s],

$\Delta x = \Delta t M_f$, $(x, q) = (0.28\ [\text{m/s}], 1)$ and $t \in [0, t_f]$. The optimal strategy is to reach the highest gear as fast as possible and then stabilize at a value $u \approx 0.5$. A different scenario is shown in Fig. 3.5.4, in which we set the initial state to $(x, q) = (14.58\ [\text{m/s}], 1)$. Here, the control lets the vehicle slow down, then switches to the third gear in order to replicate the previous strategy.

3.5.3 Bang–Bang control of a chemotherapy model

In this test, we consider the control of a two-compartment model of tumor growth. For this model, and cost functionals of the kind we will consider below, optimal controls are known to be of bang–bang type (see [42]). In this case, we can recast the problem in hybrid form, by considering an evolution in lack of chemotherapy ($Q = 1$):

$$\begin{cases} \dot{X}_1(t) = -a_1 X_1(t) + 2a_2 X_2(t) \\ \dot{X}_2(t) = a_1 X_1(t) - a_2 X_2(t) \end{cases} \tag{3.5.1}$$

and a different evolution at full-dose chemotherapy ($Q = 2$):

$$\begin{cases} \dot{X}_1(t) = -a_1 X_1(t) \\ \dot{X}_2(t) = a_1 X_1(t) - a_2 X_2(t). \end{cases} \tag{3.5.2}$$

Here, the two compartments represent the number of cells at different stages of their lives, and the chemotherapy acts by preventing the generation of new tumor cells in the first compartment by inhibiting the mitosis of cells in the second compartment.

The cost functional is defined as

$$J(x, q; \theta) = \int_0^\infty (r_1 \dot{X}_1(t) + r_2 \dot{X}_2(t) + Q(t) - 1) e^{-\lambda t} dt \tag{3.5.3}$$

in which $\dot{X}_1(t)$ and $\dot{X}_2(t)$ are given by (3.5.1)–(3.5.2) for respectively $Q(t) = 1$ and $Q(t) = 2$, and we have to minimize a combination between the growth of the tumor mass and the toxic effect of the drug on healthy cells (note that this latter term appears only when $Q(t) = 2$). Due to the geometric properties of the problem, Zeno executions cannot occur, and we can avoid to introduce a switching cost, which would have no practical meaning. Setting the switching cost to zero also causes the two value functions to coincide, i.e., $V(x, 1) \equiv V(x, 2)$, and in this case a switch can occur at $t = 0^+$. While the general theory usually rules out this situation, no particular problems arise in this specific case.

The values of the parameters are assigned as in Table 3.5.5, according to the current literature (see [42]). Figg. 3.5.5–3.5.7 show the value function(s) of the problem, the optimal switching with respect to time and space and a

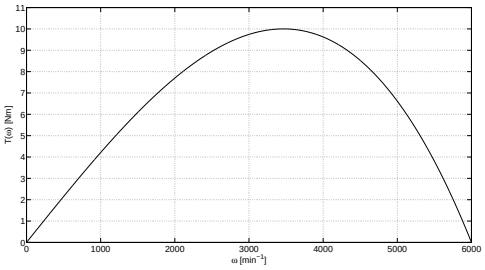


Figure 3.5.2: Power band of the engine.

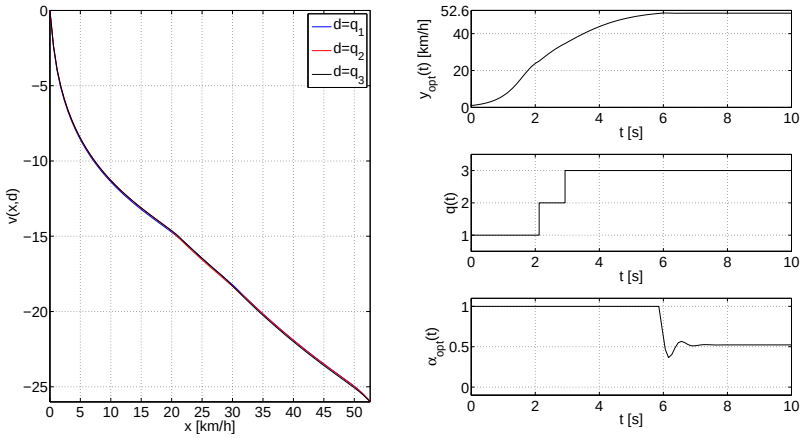


Figure 3.5.3: Value functions, trajectory and optimal control, three-gear vehicle, first case.

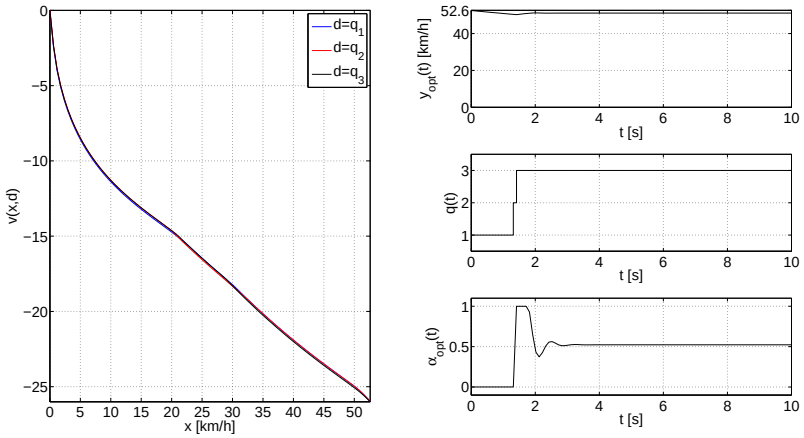


Figure 3.5.4: Value functions, trajectory and optimal control, three-gear vehicle, second case.

a_1	a_2	r_1	r_2	λ	x_1	x_2
0.197	0.356	6.94	3.94	0.1	2	1

Table 3.5.5: Choice of parameters, chemotherapy test

ϵ	N_V	N_P
10^{-3}	192	192
10^{-6}	528	526

Table 3.5.6: Number of iterations for a given tolerance ϵ, chemotherapy test.

sample trajectory starting from the initial state $(x_1, x_2) = (2, 1)$. The value function has been computed with a 100×100 grid on the domain $[0, 2]^2$, and $\Delta t = 0.1$. Note that there exists a clear discontinuity for the gradient of the value function, which corresponds to the switching curve in Fig. 3.5.7, which separates the black region, in which the optimal solution is $Q(t) = 1$, from the white region, in which the optimal solution is $Q(t) = 2$. The approximate optimal control shows a limit cycle in which a quasi-periodic switching between the two dynamics takes place.

Table 3.5.6 compares the two (VI and MPI) numerical solvers. Here and in the following test, the MPI algorithm has been implemented with $N_{\text{it}} = 10$, and an initial block of 10 value iterations has been performed at the very start in order to provide a better initial guess. As remarked above, the Modified Policy Iteration algorithm performs essentially the same number of iterations than the value iteration algorithm, but at a lower cost.

3.5.4 DC/AC inverter

The last test presents a single-phase DC/AC inverter, whose conceptual structure is sketched in Fig. 3.5.8.

In this device, a DC source generates an AC output by means of a suitable operation of the switches $S_1, \dots, S_4$, as well as a suitable choice of the three components (R, L and C) which appear in series in the RLC load. Following [20], we consider as state variables $X_1 = i_L$ (the current through the inductor L, i.e., through the load) and $X_2 = v_C$ (the voltage across the capacitor C), the state equations being

$$\begin{cases} \dot{X}_1(t) = \dfrac{V_{DC}}{L}(Q(t) - 2) - \dfrac{R}{L}X_1(t) - \dfrac{1}{L}X_2(t) \\ \dot{X}_2(t) = \dfrac{1}{C}X_1(t). \end{cases} \tag{3.5.4}$$

The physical meaning of the discrete state variable depends on the state of

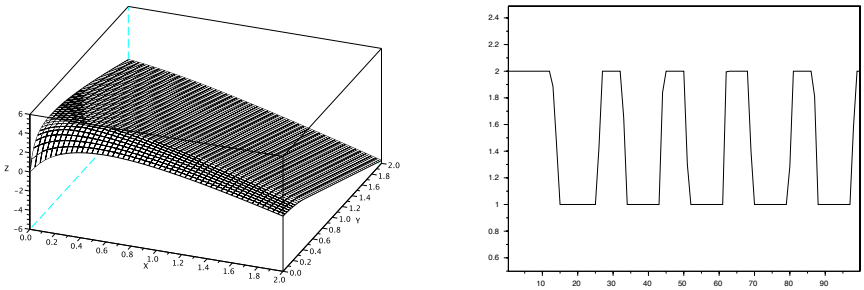


Figure 3.5.5: Value function and optimal switching for the chemotherapy test.

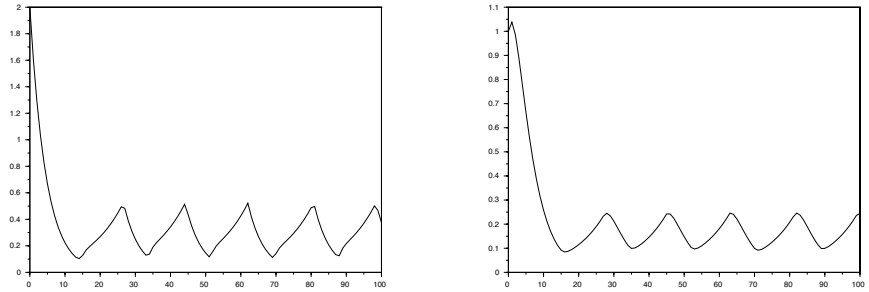


Figure 3.5.6: Trajectories $X_1(t)$ and $X_2(t)$ for the chemotherapy test.

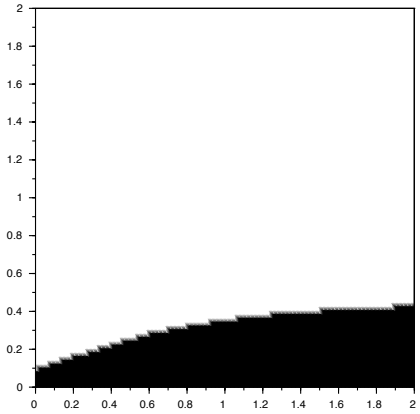


Figure 3.5.7: Optimal switching map for the chemotherapy test.

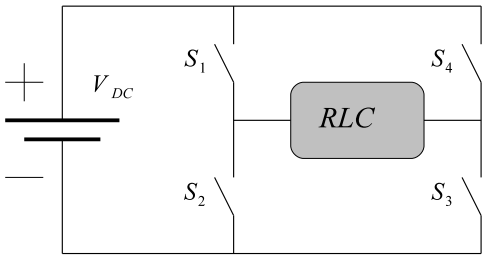


Figure 3.5.8: Abstract structure of the single-phase DC/AC inverter.

the switches $S_1, \dots, S_4$, and more precisely

$$Q(t) = \begin{cases} 1 & \text{if } S_1, S_3 = OFF \text{ and } S_2, S_4 = ON \\ 2 & \text{if } S_1, S_4 = OFF \text{ and } S_2, S_3 = ON \\ 3 & \text{if } S_2, S_4 = OFF \text{ and } S_1, S_3 = ON. \end{cases}$$

The cost functional is defined so as to force the system to evolve (approximately) along an ellipse of the state space (see [20]), namely

$$\frac{x_1^2}{a^2} + \frac{x_2^2}{b^2} = c$$

in which the constants a and b are defined in terms of the physical parameters R, L, C and of the desired pulsation ω. This makes it natural to define the running cost as

$$\ell(x, q, u) = \left(\frac{x_1^2}{a^2} + \frac{x_2^2}{b^2} - c \right)^2 . \tag{3.5.5}$$

With the parameters chosen (see Table 3.5.7), $a \approx 0.84$, $b \approx 1.34$ and the required output of the system would be given by two sinusoids of amplitude respectively 126 A for X_1 and 200 V for X_2, both at the frequency of 1 Hz. The approximate solution has been computed on a 100×100 grid on the domain $[-250, 250]^2$, with $\Delta t = 0.01$, and state constraint boundary conditions have been treated by penalization, assigning a stopping cost of $5 \cdot 10^8$ on the boundary. The effect of the lack of full controllability is apparent in Fig. 3.5.9, which shows one component of the value function (they are practically undistinguishable from one another) and the optimal switching with respect to time. Fig. 3.5.10 shows the output $(X_1(t), X_2(t))$ of the controlled system, whereas, as an example, Fig. 3.5.11 reports the switching map of the second component of the state space. Here, the optimal solution is to keep $Q(t) = 2$ in grey regions, commute to $Q(t) = 1$ in black regions and to $Q(t) = 3$ in white regions.

Finally, Table 3.5.8 compares the two numerical solvers (VI and MPI). In this last test, the stopping condition has been computed on the relative

V_{DC}	R	L	C	ω	c	λ	x_1	x_2
200 [V]	0.7 [Ω]	0.1 [H]	0.1 [F]	2π [s^{-1}]	22500	1	0	200

Table 3.5.7: Choice of parameters, inverter

ϵ	N_V	N_P
10^{-3}	469	469
10^{-6}	13481	13491

Table 3.5.8: Number of iterations for a given tolerance ϵ, DC/AC inverter test.

l^1 update,

$$\frac{|\boldsymbol{v}_j - \boldsymbol{v}_{j-1}|_{l^1}}{|\boldsymbol{v}_j|_{l^1}} < \epsilon$$

to avoid problems with both high values of the solution and the occurrence of a discontinuity caused by the lack of controllability.

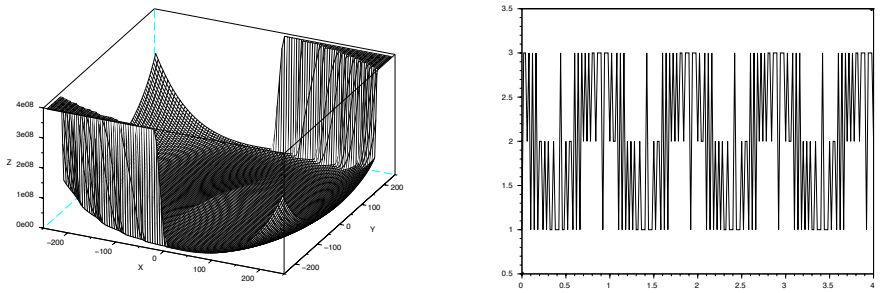


Figure 3.5.9: Value function and optimal switching for the DC/AC inverter.

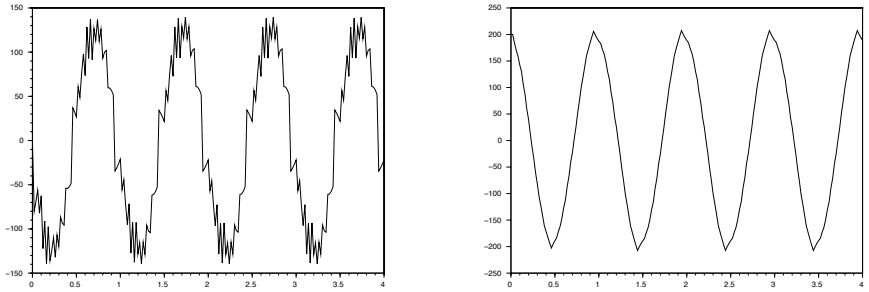


Figure 3.5.10: Trajectories $X_1(t)$ and $X_2(t)$ for the DC/AC inverter.

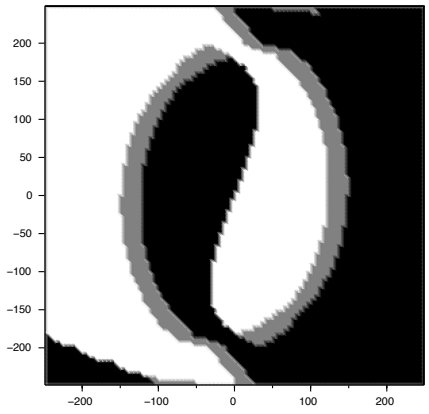


Figure 3.5.11: Optimal switching map for $q = 2$ for the DC/AC inverter.

Chapter 4

Error estimates for the numerical scheme

The main goal of this chapter is to derive error estimate between the value function V and its approximation V_h. The arguments that will be used here are mainly based on viscosity notion and on the equations satisfied by V and V_h. The proof follows some ideas introduced in [38, 39] and it is based on the perturbation and shaking coefficients technique, relying on some sensitivity analysis of the value function with respect to perturbations of the trajectories.

Remark 4.0.1. *Throughout this chapter we will study the problem in the particular case where optional transitions are always allowed and mandatory transitions are always forbidden, i.e.* $\mathcal{C} = \mathbb{R}^d \times \mathbb{I}$ *and* $\mathcal{A} = \emptyset$.

Remark 4.0.2. *Here we analyze a general form of the numerical scheme* S *approximating* (2.2.13)*, assuming that it has a solution* V_h*. We verify in Appendix A.2 that this holds for the Semi-Lagrangian scheme, but the proof can be extended to Finite Differences schemes.*

In order to prove the results of this chapter, we will make use of two additional assumptions.

(A8) The discount factor λ satisfies both $\lambda > 1$ and $\lambda > L_f$.

This assumption is convenient for the computation of the estimates, but it's not strictly necessary as suggested by the positive results of the numerical tests of the previous chapter.

(S4) Let $\eta \geq 0$ be a constant. If v is solution of

$$\max\Big\{S(h,x,q,v(x,q),v), v(x,q) - \mathcal{N}v(x,q)\Big\} = 0$$

then $v + \eta$ is solution of

$$\max\Big\{S(h,x,q,v(x,q),v) + \eta\lambda; v(x,q) - \mathcal{N}v(x,q)\Big\} = 0$$

Moreover, if S can be written in the form

$$S(h,x,q,r,\phi) = \max_{u\in U} S^u(h,x,q,r,\phi)$$

then for $\mu \in (0,1)$, μv is sub-solution of

$$\begin{aligned}\max\Big\{ &\max_{u\in U} S^u(h,x,q,\mu v(x,q) + (\mu-1)\ell(x,q,u), \mu v),\\ &\mu v(x,q) - \mu\mathcal{N}v(x,q)\Big\} \le 0\end{aligned} \tag{4.0.1}$$

4.1 Cascade Problems

The main difficulties in this study come from the presence of controlled jumps which leads to an HJB equation with coupling terms that involve the highly non-linear operator $\mathcal{N}$. To deal with these difficulties, we use the idea of *cascade problems* that we describe in the following subsections.

4.1.1 Cascade for the HJB equation

We approach equation (2.2.13) with a sequence of obstacle problems and use the same methods as in [36, Proof of Theorem 4.2] to prove that the solutions of the sequence of equations converges to the solution of (2.2.13).

Consider the following problem:

$$\lambda V_0(x,q) + H(x,q,D_x V_0(x,q)) = 0 \quad \text{on } \mathbb{R}^d \times \mathbb{I}. \tag{4.1.1}$$

Under assumptions (A1)-(A2), this equation has a unique viscosity solution V_0 in $C_{\mathrm{b,l}}(\mathbb{R}^d \times \mathbb{I})$. Since $V \equiv 0$ is a viscosity sub-solution of (4.1.1), the comparison principle (see [36, Theorem 3.3]) implies $0 \le V_0$. Now, for a given V_{n-1} in $C_{\mathrm{b,l}}(\mathbb{R}^d \times \mathbb{I})$ and $n \ge 1$, consider the problem

$$\begin{aligned}\max\Big\{&\lambda V_n(x,q) + H(x,q,D_x V_n(x,q)),\\ &V_n(x,q) - \mathcal{N}V_{n-1}(x,q)\Big\} = 0\end{aligned} \quad \text{on } \mathbb{R}^d \times \mathbb{I}. \tag{4.1.2}$$

Since $\mathcal{N}V_{n-1}$ is uniformly continuous, under assumptions (A1)-(A2), there exists a unique viscosity solution V_n of (4.1.2) in $C_{\mathrm{b,l}}(\mathbb{R}^d \times \mathbb{I})$. It is easy to check that V_1 is a viscosity sub-solution of (4.1.1). By the comparison principle, $V_1 \le V_0$. Moreover, $V \equiv 0$ is a sub-solution of (4.1.2) for $n = 1$, and then $0 \le V_1 \le V_0$ in $\mathbb{R}^d$. By point (1) of Proposition 2.2.1 $\mathcal{N}V_1 \le \mathcal{N}V_0$, then we can say that V_2 is a viscosity sub-solution of (4.1.2) for $n = 1$, and also $V_2 \le V_1$ in $\mathbb{R}^d$.

By induction over n, we obtain:

$$0 \le \cdots \le V_n \le \cdots \le V_2 \le V_1 \le V_0. \tag{4.1.3}$$

We can see that, if $|V_0|_0 \leq K_0$ (where K_0 is defined in assumption (A7)), then $V = V_0$ is a viscosity solution of (4.1.1) (we refer to Section 4.3.2 for error estimates).

Suppose now that $|V_0|_0 > K_0$, and let $\mu \in (0,1)$ such that $\mu|V_0|_0 < K_0$.

Theorem 4.1.1. *We have that, for all n,*

$$V_n - V_{n+1} \leq (1-\mu)^n |V_0|_0. \tag{4.1.4}$$

Moreover, V_n converges towards V, when n tends to $+\infty$ and

$$0 \leq V_n - V \leq \frac{(1-\mu)^n}{\mu} |V_0|_0. \tag{4.1.5}$$

Proof. The same arguments used in [36, Proof of Theorem 4.2] can be used here. For convenience of the reader, we give here the main steps. Let $n \in \mathbb{N}$, and $\theta_n \in (0,1]$ be such that, in $\mathbb{R}^d \times \mathbb{I}$

$$V_n - V_{n+1} \leq \theta_n V_n. \tag{4.1.6}$$

By (4.1.3), this holds at least for $\theta_n = 1$. Rewriting (4.1.6) as $(1-\theta_n)V_n \leq V_{n+1}$, and using Proposition 2.2.1, get

$$(1-\theta_n)\mathcal{N}V_n + \theta_n K_0 \leq (1-\theta_n)\mathcal{N}V_n + \theta_n \mathcal{N}0 \leq \mathcal{N}\big((1-\theta_n)V_n\big) \leq \mathcal{N}V_{n+1}. \tag{4.1.7}$$

We now prove that

$$(1-\theta_n+\mu\theta_n)V_{n+1} \leq V_{n+2} \tag{4.1.8}$$

where V_{n+2} is the viscosity solution of (4.1.2) with $n+2$. Since V_{n+1} is the viscosity solution of (4.1.2) with $n+1$, and $\ell(x,q,u) \geq 0$, for all x, for all q, and for all u, we have that $(1-\theta_n+\mu\theta_n)V_{n+1}$ is a viscosity sub-solution of

$$\lambda v(x,q) + H(x,q,D_x v(x,q)) \leq 0 \qquad (x,q) \in (\mathbb{R}^d \times \mathbb{I}) \setminus \mathcal{C}.$$

Moreover, by the construction of the sequence (4.1.3), and by (4.1.7), we have

$$(1-\theta_n+\mu\theta_n)V_{n+1} \leq (1-\theta_n)V_{n+1} + \mu\theta_n|V_0|_0 \tag{4.1.9}$$

$$\mathcal{N}V_{n+1} \geq (1-\theta_n)\mathcal{N}V_n + \theta_n K_0. \tag{4.1.10}$$

Taking the difference between (4.1.9) and (4.1.10), and knowing that V_{n+1} is the viscosity solution of (4.1.2), we obtain

$$\begin{aligned}
&(1-\theta_n+\mu\theta_n)V_{n+1} - \mathcal{N}V_{n+1} \leq \\
&\quad \leq (1-\theta_n)V_{n+1} + \mu\theta_n|V_0|_0 - (1-\theta_n)\mathcal{N}V_n - \theta_n K_0 \leq \\
&\quad \leq (1-\theta_n)V_{n+1} + \theta_n K_0 - (1-\theta_n)\mathcal{N}V_n - \theta_n K_0 \leq 0.
\end{aligned}$$

The same arguments also lead to

$$(1-\theta_n+\mu\theta_n)V_{n+1}-\mathcal{N}V_{n+1}\leq 0.$$

So we can say that $(1-\theta_n+\mu\theta_n)V_{n+1}$ is a viscosity sub-solution of (4.1.2) with $n+2$. The comparison principle implies (4.1.8), or equivalently

$$V_{n+1}-V_{n+2}\leq \theta_n(1-\mu)V_{n+1}. \tag{4.1.11}$$

By the inequalities $V_0-V_1\leq V_0$ in $\mathbb{R}^d\times\mathbb{I}$, we obtain $V_1-V_2\leq(1-\mu)V_1$ in $\mathbb{R}^d\times\mathbb{I}$. Then, taking $\theta_1=1-\mu$ yields to $V_2-V_3\leq(1-\mu)^2V_2$, and by induction we have

$$V_{n+1}-V_{n+2}\leq(1-\mu)^{n+1}V_{n+1}\leq(1-\mu)^{n+1}|V_0|_0. \tag{4.1.12}$$

By (4.1.3) and (4.1.4), we can find a function $V\in C(\mathbb{R}^d\times\mathbb{I})$, such that $|V_n-V|_0\to 0$, when $n\to+\infty$. Proposition 2.2.1 and the stability of solutions imply that V is a viscosity solution of (2.2.13). Then we can say that V_n converges to V, the unique viscosity solution of (2.2.13), when $n\to+\infty$. Moreover, by (4.1.4) and since $(1-\mu)<1$, the following upper bound holds in $\mathbb{R}^d\times\mathbb{I}$ for all $n\geq 0$

$$0\leq V_n-V\leq\sum_{i=n}^{+\infty}(1-\mu)^i|V_0|_0=\frac{(1-\mu)^n}{1-(1-\mu)}|V_0|_0=\frac{(1-\mu)^n}{\mu}|V_0|_0. \tag{4.1.13}$$

□

4.1.2 Cascade for the numerical scheme

As we have done for the equation (2.2.13), we will approach (3.1.1) by a sequence of equations approximating (4.1.2).

Let $V_{h0}\in C_{\mathrm{b}}(\mathbb{R}^d\times\mathbb{I})$ be a solution of

$$S\big(h,x,q,V_{h0}(x,q),V_{h0}\big)=0\quad\text{on }\mathbb{R}^d\times\mathbb{I} \tag{4.1.14}$$

Define $V_{h1}\in C_{\mathrm{b}}(\mathbb{R}^d\times\mathbb{I})$ a solution of the problem:

$$\max\Big\{S\big(h,x,q,v(x,q),v\big),v(x,q)-\mathcal{N}V_{h0}(x,q)\Big\}=0\quad\text{on }\mathbb{R}^d\times\mathbb{I}. \tag{4.1.15}$$

For $n=2,3,\cdots$, we suppose that there exists a continuous and bounded solution V_{hn} of

$$\max\Big\{S\big(h,x,q,v(x,q),v\big),v(x,q)-\mathcal{N}V_{h(n-1)}(x,q)\Big\}=0\quad\text{on }\mathbb{R}^d\times\mathbb{I}. \tag{4.1.16}$$

Again, as pointed out in Remark 4.0.2, the fact that V_{hn} exists will be proven in the appendix for the particular case of the Semi-Lagrangian scheme.

The function V_{h1} is a sub-solution of (4.1.14), and then $V_{h1} \leq V_{h0}$ in $\mathbb{R}^d \times \mathbb{I}$. Using proposition 2.2.1 and assumption (S4), one can verify that $V_h \equiv 0$ is a sub-solution of (4.1.15) in $\mathbb{R}^d \times \mathbb{I}$, which gives that $0 \leq V_{h1} \leq V_{h0}$ in $\mathbb{R}^d \times \mathbb{I}$. Proposition 2.2.1 implies that $0 \leq \mathcal{N}V_{h1} \leq \mathcal{N}V_{h0}$, then V_{h2} is a sub-solution of (4.1.15), and hence $V_{h2} \leq V_{h1}$ in $\mathbb{R}^d \times \mathbb{I}$. By induction on n, it follows

$$0 \leq \cdots \leq V_{hn} \leq \cdots \leq V_{h2} \leq V_{h1} \leq V_{h0} \tag{4.1.17}$$

As in Subsection 4.1.1, we suppose that $|V_0|_0 > K_0$. Then, since $V_{h0} \to V_0$ uniformly (Theorem 3.1.1), we have also $|V_{h0}|_0 > K_0$ for h small enough and we can choose $\mu \in (0,1)$ such that $\mu|V_0|_0 < K_0$, and $\mu|V_{h0}|_0 < K_0$.

If the scheme S is such that V_{hn} and V_h exist, then the following convergence result holds:

Theorem 4.1.2. *Suppose that, for every n, problems* (4.1.14)-(4.1.16) *admit solution. Then for all n and for h small enough, in $\mathbb{R}^d \times \mathbb{I}$ we have*

$$V_{hn} - V_{h(n+1)} \leq (1-\mu)^n |V_{h0}|_0 \tag{4.1.18}$$

Proof. We use the same methods as in Theorem 4.1.1, taking into account the monotonicity of S. □

Proposition 4.1.1. *Under assumptions (S1)-(S4), if the equation* (4.1.16) *admits a unique solution V_{hn} for every $n \geq 2$, we have $|V_{hn} - V_h|_0 \to 0$ when $n \to +\infty$ and, on $\mathbb{R}^d \times \mathbb{I}$,*

$$V_{hn} - V_h \leq \sum_{i=n}^{+\infty} (1-\mu)^i |V_{h0}|_0 = \frac{(1-\mu)^n}{\mu} |V_{h0}|_0 \quad \forall n \geq 1. \tag{4.1.19}$$

Proof. By (4.1.17) and (4.1.18), we can find a function $V_h \in C_{\mathrm{b}}(\mathbb{R}^d \times \mathbb{I})$, such that $|V_{hn} - V_h|_0 \to 0$, when $n \to +\infty$. The stability property of solutions implies that V_h is a solution of (3.1.1). □

If (4.1.14), (4.1.15) and (4.1.16) admit solutions V_{hn}, then they converge towards the solutions V_n of (4.1.1) and (4.1.2) and we also have (4.1.17) and (4.1.19).

Moreover, we assume that V_{hn} is Lipschitz continuous for every $n \geq 0$ and that

$$0 \leq \cdots \leq L_{V_{hn}} \leq \cdots \leq L_{V_{h2}} \leq L_{V_{h1}} \leq L_{V_{h0}} \tag{4.1.20}$$

We verify in the appendix that this holds for the Semi-Lagrangian scheme.

4.2 Lipschitz continuity

We point out that, in order to establish the approximation error of the scheme, we need V to be Lipschitz (or at least Hölder) continuous. In general, problem (2.2.13) is expected to have a Hölder continuous solution (see

[26]). However, assumption (A4) ensures non-expansivity of the Lipschitz constants by the jump operator $\mathcal{N}$. Therefore, for λ large enough it is possible to prove that the value function is Lipschitz continuous. This claim is stated precisely and proved in this section.

Lemma 4.2.1. *Under assumption (A1)-(A8), the viscosity solution V_0 of the HJB equation* (4.1.1) *is Lipschitz continuous and its Lipschitz constant is given by:*

$$L_{V_0} = \frac{L_\ell}{\lambda - L_f}.$$

Proof. This is a classical result and its proof can be found in [3]. □

Now, consider a general HJB equation of the form:

$$\max\Big\{\lambda w(x,q) + H(x,q,D_x w(x,q)), w(x,q) - \Phi(x,q)\Big\} = 0 \quad \text{on } \mathbb{R}^d \times \mathbb{I}. \tag{4.2.1}$$

where $\Phi : \mathcal{C} \to \mathbb{R}$ is Lipschitz continuous. Again, by using classical arguments in viscosity theory, we get the following lemma.

Lemma 4.2.2. *Under assumptions (A1)-(A8), equation* (4.2.1) *admits a unique bounded Lipschitz continuous viscosity solution w. Moreover, the Lipschitz constant of w satisfies:*

$$L_w = \max\left\{L_\Phi, \frac{L_\ell}{\lambda - L_f}\right\}$$

Proof. The proof can be found in Appendix A.1. □

This Lemma 4.2.2 and the cascade construction, lead directly to the following conclusion.

Theorem 4.2.1. *Assume (A1)-(A8). The value function V is Lipschitz continuous and an upper bound of its Lipschitz constant is L_{V_0}:*

$$|V|_1 \leq L_{V_0}$$

Proof. Consider the cascade construction and the associated sequence $\{V_n\}$. We claim that for any $n \geq 0$, an upper bound of the Lipschitz constant of V_n is given by:

$$L_{V_n} = \max\{L_{V_0}, L_{c\mathcal{C}}\} \tag{4.2.2}$$

For $n = 0$, this result is stated in Lemma 4.2.1. Now, assume that (4.2.2) holds for $n \geq 0$ and let us prove that the statement remains valid for $n+1$.

First, notice that for hypothesis (A7), for every x_1 and x_2

$$\begin{aligned}|\mathcal{N}V_n(x_1,q)-\mathcal{N}V_n(x_2,q)| \leq \Big| &\min_{(x',q')\in\mathcal{D}}\{V_n(x',q')+c_{\mathcal{A}}(x_1,q,x',q')\}- \\ &-\min_{(x',q')\in\mathcal{D}}\{V_n(x',q')+c_{\mathcal{A}}(x_2,q,x',q')\}\Big| \leq \\ \leq \Big| &\sup_{(x',q')\in\mathcal{D}}\{c_{\mathcal{A}}(x_1,q,x',q')- \\ &-c_{\mathcal{A}}(x_2,q,x',q')\}\Big| \leq \\ \leq\ & L_{c_{\mathcal{C}}}|x_1-x_2|.\end{aligned}$$

Hence, by combining the previous inequality with Lemma 4.2.2, we deduce that

$$|V_{n+1}|_1 \leq \frac{L_\ell}{\lambda - L_f}.$$

Using (4.1.3), we conclude that an upper bound of $|V_{n+1}|_1$ is $L_{V_{n+1}} = L_{V_0}$ which ends the proof. □

4.3 Error estimates

Before starting the analysis of error estimates for the approximation of (2.2.13), we first analyze two intermediate problems. The first one corresponds to the first iteration in the cascade problems defined in the previous section.

4.3.1 The Hamilton-Jacobi equation with obstacles

Consider first the viscosity solution w of the general HJB equation (4.2.1) and define an approximation w_h of w as solution of the following numerical scheme:

$$\max\Big\{S(x,q,w_h(x,q),w_h),w_h(x,q)-\Phi(x,q)\Big\}=0 \quad \text{on } \mathbb{R}^d\times\mathbb{I}. \tag{4.3.1}$$

In the sequel, we assume that for every $h>0$, equation (4.3.1) admits a solution w_h, and in Appendix A.2 we show that this requirement is valid at least for a SL scheme.

We want to analyze the error estimate between w and w_h for a small mesh size h. Unfortunately, due to the non-linearity of the obstacles $\mathcal{N}V_{n-1}$ and $\mathcal{N}V_{h(n-1)}$ appearing in (4.1.2) and (4.1.16), the classical approach developed by Capuzzo Dolcetta and Souganidis cannot be applied to our problem. For this reason, the arguments that used here to obtain the error estimates are based on the shaking coefficients and regularization method introduced by

Krylov in [38, 39]. To use this method, some further notations are needed. Consider a sequence of mollifiers $\{\rho_\varepsilon\}$ defined as follows:

$$\rho_\varepsilon(x) = \varepsilon^{-d}\rho\left(\frac{x}{\varepsilon}\right) \tag{4.3.2}$$

where $\rho \in C^\infty(\mathbb{R}^d)$, $\int_{\mathbb{R}^d}\rho = 1$, $\text{supp}\{\rho\} \subseteq \bar{B}(0,1)$ and $\rho \geq 0$. We define the mollification of $\phi \in C_b(\mathbb{R}^d)$ as follows:

$$\phi_\varepsilon(x) := \phi * \rho = \int_{\mathbb{R}^d}\phi(x-e)\rho_\varepsilon(e)de. \tag{4.3.3}$$

If ϕ is Lipschitz continuous, then

$$|\phi(x) - \phi_\varepsilon(x)| \leq L_\phi\varepsilon, \quad \text{and} \quad |D^i\phi_\varepsilon(x)| \leq L_\phi\varepsilon^{1-i}|\phi|_0 \tag{4.3.4}$$

Lemma 4.3.1. *Assume (A1)-(A8). For every $0 < \varepsilon < \frac{\beta}{4}$ (where β is as in (A2)), the following assertions hold:*

i) There is a unique solution w^ε of

$$\begin{aligned}\max\Big\{&\lambda w^\varepsilon(x,q) + \max_{|e|\leq\varepsilon} H\big(x+e,q,D_x w^\varepsilon(x,q)\big),\\ &w^\varepsilon(x,q) - \Phi(x,q)\Big\} = 0\end{aligned} \qquad \text{on } \mathbb{R}^d \times \mathbb{I}. \tag{4.3.5}$$

ii) The following estimate holds:

$$|w - w^\varepsilon|_0 \leq \varepsilon K_\Phi$$

where w is a solution of (4.2.1) and $K_\Phi := \frac{L_\ell}{\lambda - L_f} + L_\Phi\frac{L_f}{\lambda}$. Moreover, we have

$$|w^\varepsilon|_1 \leq L_w = \max\left\{L_\Phi, \frac{L\ell}{\lambda - L_f}\right\}.$$

iii) If we define $w_\varepsilon := w^\varepsilon * \rho_\varepsilon$. Then there exists $C > 0$, such that w_ε is a classical sub-solution of

$$\begin{aligned}\max\Big\{&\lambda w_\varepsilon(x,q) + H\big(x,q,D_x w_\varepsilon(x,q)\big),\\ &w_\varepsilon(x,q) - C\varepsilon - \Phi(x,q)\Big\} \leq 0\end{aligned} \qquad \text{on } \mathbb{R}^d \times \mathbb{I}. \tag{4.3.6}$$

Proof. i) The existence and uniqueness of solution w^ε is standard, as it is viscosity solution of the stopping control problem described below (we also report a more general formulation in Appendix A.3).

Let us consider the following variation of the dynamics described in (2.2.1):

$$\begin{cases} \dot{X}^\varepsilon(t) = f(X^\varepsilon(t) + e(t), Q(t), u(t)) \\ X^e(0) = x \\ Q(0^+) = q \end{cases}$$

where, given $\varepsilon > 0$, $e \in \mathcal{F}^\varepsilon$ with

$$\mathcal{F}^\varepsilon := \{e : (0, +\infty) \to \mathbb{R}^d \quad | \quad e \text{ measurable}, |e(t)| \leq \varepsilon \text{ a.e.}\}$$

With these dynamics we define a stopping control problem in which the ability to switch between dynamics is replaced by the ability to stop at any moment. The stopping time is denoted by ξ and, in case the controller doesn't choose to stop, its value is $+\infty$ by definition.

The control strategy θ^ε consists then in the continuous controls u and e and the controlled stopping time ξ (which can be finite or infinite) and belongs to the set $\Theta^\varepsilon := \mathcal{U} \times \mathbb{R}_+ \times \mathcal{F}^\varepsilon$.

From [3], we know that the function w^ε defined as

$$w^\varepsilon(x, q) := \inf_{\theta^\varepsilon \in \Theta^\varepsilon} J^\varepsilon(x, q; \theta^\varepsilon)$$

where

$$J^\varepsilon(x, q; \theta^\varepsilon) := \int_0^\xi \ell(X_x^\varepsilon(t, q, u) + e(t), q, u(t)) e^{-\lambda t} dt + \\ + e^{-\lambda \xi} \Phi(X_x^\varepsilon(\xi, q, u), q).$$

is solution of (4.3.5).

ii) The stability result is also proved in Appendix A.3, while the estimate on the Lipshitz constant of w^ε is obtained in Appendix A.1.

iii) First, note that w^ε is subsolution of the equation:

$$\lambda w^\varepsilon(x, q) + \max_{|e| \leq \varepsilon} H(x + e, q, D_x w^\varepsilon(x, q)) \leq 0 \quad (x, q) \in \mathbb{R}^d \times \mathbb{I}.$$

By a straightforward adaptation of the arguments in [5, Lemma A3], we prove that w_ε is a subsolution of

$$\lambda w_\varepsilon(x, q) + H(x, q, D_x w_\varepsilon(x, q)) \leq 0 \quad (x, q) \in \mathbb{R}^d \times \mathbb{I}.$$

Moreover, since $w^\varepsilon \leq \Phi$ on $\mathcal{C}$ and w^ε and Φ are Lipschitz continuous, for any $x \in \mathcal{C}$ we have:

$$w_\varepsilon(x, q) := \int_{|e| \leq 1} w^\varepsilon(x - \varepsilon e, q) \rho(e) de \leq \\ \leq \int_{|e| \leq 1} w^\varepsilon(x, q) \rho(e) de + L_w \varepsilon \leq \\ \leq \Phi(x, q) + L_w \varepsilon.$$

□

The same result holds also for the scheme. Indeed, one can define the perturbed scheme by:

$$\max\Big\{\max_{|e|\leq\varepsilon} S(x+e,q,w_h^\varepsilon(x,q)), w_h^\varepsilon(x,q)-\Phi(x,q)\Big\}=0 \quad \text{on } \mathbb{R}^d\times\mathbb{I}. \tag{4.3.7}$$

If we assume that this scheme has a solution w_h^ε (see Remark 4.0.2), then the following statement holds.

Lemma 4.3.2. *Let $0<\varepsilon<\frac{\beta}{4}$ (where β is as in (A2)). If we define $w_{h,\varepsilon}:=w_h^\varepsilon * \rho_\varepsilon$. Then there exists a constant $C>0$ such that $w_{h,\varepsilon}$ is a classical subsolution of*

$$\max\Big\{S(x,q,w_{h,\varepsilon}(x,q),w_{h,\varepsilon}), w_{h,\varepsilon}(x,q)-\Phi(x,q)-C\varepsilon\Big\}\leq 0 \quad \text{on } \mathbb{R}^d\times\mathbb{I}. \tag{4.3.8}$$

Proof. This result is derived with the same arguments as in the proof of Lemma 4.3.1(iii). □

In addition, we shall assume that the following estimates hold:

$$\begin{aligned} &|w_h|_1\leq L_{w_h}:=(1+(\lambda-L_f)h)\max\left\{L_\Phi,\frac{L_\ell}{\lambda-L_f}\right\}\\ &|w_h^\varepsilon|_1\leq L_{w_h} \end{aligned} \tag{4.3.9}$$

and

$$|w_h-w_h^\varepsilon|_0\leq \varepsilon K_{w_h,h} \tag{4.3.10}$$

where w_h is a solution of (4.3.1) and

$$K_{w_h,h}:=\max\left\{(L_\ell+L_{w_h}L_f)h,\frac{L_\ell+L_{w_h}L_f}{\lambda}\right\}.$$

These assumptions are proved in Sections A.2 and A.3.4 for the case of a monotone Semi-Lagrangian scheme. The same arguments can be used for other classical monotone schemes.

Lastly, we assume that, for an obstacle function $\tilde\Phi$, the solution $\tilde w_h$ of

$$\max\Big\{S(x,q,\tilde w_h(x,q),\tilde w_h), \tilde w_h(x,q)-\tilde\Phi(x,q)\Big\}=0 \quad \text{on } \mathbb{R}^d\times\mathbb{I}$$

and the solution w_h of (4.3.1) satisfy

$$|w_h(x,q)-\tilde w_h(x,q)|\leq|\Phi(x,q)-\tilde\Phi(x,q)| \quad \forall (x,q)\in\mathbb{R}^d\times\mathbb{I}. \tag{4.3.11}$$

The previous result can be easily obtained for SL schemes.

Proposition 4.3.1. *Assume (A1)-(A8) and (S1)-(S4). If problem* (4.3.1) *and* (4.3.7) *admit solutions and* (4.3.9) *and* (4.3.10) *hold, then for every* $(x,q) \in \mathbb{R}^d \times \mathbb{I}$*, we have*

$$-\underline{K}_{w_h,h}|h|^\gamma \leq w(x,q) - w_h(x,q) \leq \overline{K}_{w,\Phi}|h|^\gamma$$

where

$$\begin{aligned}\overline{K}_{w,\Phi} &:= K_\Phi + L_w + K_c L_w |w|_0 |J| \\ \underline{K}_{w_h,h} &:= K_{w_h,h} + L_{w_h} + K_c L_{w_h} |w_h|_0 |J|\end{aligned}$$

and

$$\gamma := \min_{i \in J} \frac{k_i}{i} \tag{4.3.12}$$

according to the definitions in (S3) and Lemma 4.3.1(ii).

Proof. By Lemma 4.3.1 (iii), w_ε is a classical sub-solution of (4.3.6). Therefore, by (S3) and (4.3.4), we have:

$$\begin{aligned}S(h,x,q,w_\varepsilon(x,q),w_\varepsilon) &\leq \lambda w_\varepsilon(x,q) + H\big(x,q,Dw_\varepsilon(x,q)\big) + K_c\mathcal{E}(h,w_\varepsilon) \leq \\ &\leq K_c \sum_{i\in J} |D^i w_\varepsilon|_0 |h|^{k_i} \leq K_c \sum_{i\in J} L_w \varepsilon^{1-i} |w|_0 |h|^{k_i} \leq \\ &\leq K_c L_w |w|_0 \sum_{i\in J} \varepsilon^{1-i} |h|^{k_i}.\end{aligned}$$

By comparison principle of the scheme, we get:

$$w_\varepsilon - w_h \leq K_c L_w |w|_0 \sum_{i\in J} \varepsilon^{1-i} |h|^{k_i}.$$

In order to determine γ, we substitute $\varepsilon = |h|^\gamma$ in the previous estimate to obtain

$$w_\varepsilon - w_h \leq K_c L_w |w|_0 \sum_{i\in J} |h|^{\gamma(1-i)+k_i}.$$

So, by choosing $\gamma = \min_{i\in J} \frac{k_i}{i}$, we have

$$w_\varepsilon - w_h \leq K_c L_w |w|_0 |J| |h|^\gamma.$$

Now, by taking (4.3.4) and Lemma 4.3.1(ii) into account, we conclude

$$\begin{aligned}w - w_h &= w - w^\varepsilon + w^\varepsilon - w_\varepsilon + w_\varepsilon - w_h \leq \\ &\leq K_\Phi |h|^\gamma + L_w |h|^\gamma + K_c L_w |w|_0 |J| |h|^\gamma\end{aligned}$$

and therefore the upper bound in Proposition 4.3.1 is satisfied.

The lower bound on $w - w_h$ follows with symmetric arguments where a smooth sub-solution of equation (4.2.1) is constructed from the regularized

numerical scheme (4.3.8). In fact, by Lemma 4.3.2 we have that $w_{h,\varepsilon}$ is a classical sub-solution of (4.3.8), and by applying (S3) and (4.3.4) we obtain

$$\begin{aligned}
&\lambda w_{h,\varepsilon}(x,q) + H\big(x,q,Dw_{h,\varepsilon}(x,q)\big) \leq \\
&\quad \leq S\big(h,x,q,w_{h,\varepsilon}(x,q),w_{h,\varepsilon}\big) + K_{\mathrm{c}}\mathcal{E}(h,w_{h,\varepsilon}) \leq \\
&\quad \leq K_{\mathrm{c}} \sum_{i\in J} |D^i w_{h,\varepsilon}|_0 |h|^{k_i} \leq \\
&\quad \leq K_{\mathrm{c}} L_{w_h} |w_h|_0 \sum_{i\in J} \varepsilon^{1-i} |h|^{k_i}.
\end{aligned}$$

Again, by using the comparison principle and substituting $\varepsilon = |h|^\gamma$ with $\gamma = \min_{i\in J} \frac{k_i}{i}$, we get

$$w_{h,\varepsilon} - w \leq K_{\mathrm{c}} L_{w_h} |w_h|_0 \sum_{i\in J} |h|^{\gamma(1-i)+k_i} \leq K_{\mathrm{c}} L_{w_h} |w_h|_0 |J| |h|^\gamma.$$

Now, by taking (4.3.4), (4.3.10) and (4.3.9) into account, we conclude

$$\begin{aligned}
w_h - w &= w_h - w_h^\varepsilon + w_h^\varepsilon - w_{h,\varepsilon} + w_{h,\varepsilon} - w \leq \\
&\leq K_{w_h,h}|h|^\gamma + L_{w_h}|h|^\gamma + K_{\mathrm{c}} L_{w_h} |w_h|_0 |J| |h|^\gamma
\end{aligned}$$

and therefore we obtain the lower bound in Proposition 4.3.1. □

Remark 4.3.1. *The value of γ can be determined explicitly once the numerical scheme is chosen. For example, in the case of the Semi-Lagrangian scheme used in Chapter 3, we proved that the consistency property reads*

$$S(\Delta t,x,q,\phi(x),\phi) = \lambda\phi(x,q) + H\big(x,q,D\phi(x,q)\big) + \sum_{i\in\{1,2\}} |D^i\phi|_0 \Delta t.$$

Hence, by setting $h = \Delta t$ and by applying the definition $\gamma := \min_{i\in J} \frac{k_i}{i}$ with $J = \{1,2\}$, $k_1 = 1$ and $k_2 = 1$, Proposition 4.3.1 leads to the estimate

$$-\underline{K}_{w_h,h}|h|^{\frac{1}{2}} \leq w(x,q) - w_h(x,q) \leq \overline{K}_{w,\Phi}|h|^{\frac{1}{2}}$$

for the SL scheme.

4.3.2 Error estimates for the case without controlled jumps

First, consider the problem (4.1.1) and its viscosity solution $V_0 \in C_{b,l}(\mathbb{R}^d \times \mathbb{I})$.

Proposition 4.3.2. *Assume that (A1)-(A8) and (S1)-(S4) hold. Then, if $\lambda > 1$, we have*

$$\big|V_0(x,q) - V_{h0}(x,q)\big| \leq C_0|h|^\gamma \quad \forall (x,q) \in \mathbb{R}^d \times \mathbb{I}$$

where

$$C_0 := 2K_c|J| \max\{L_{V_0}|V_0|_0, L_{V_{h0}}|V_{h0}|_0\}$$

and $\gamma := \min_{i\in J} \frac{k_i}{i}$, according to the definitions in (S3) and Lemma 4.3.1(ii).

Proof. This is a classical result, we report a sketch of the proof given in [17].

Take $q \in \mathbb{I}$. Let $\epsilon \in (0,1)$ and, for any (x,y) inside the set $\mathbb{R}^d \times \mathbb{R}^d$, consider the test function

$$\Psi_\epsilon(x,y) := V_{h0}(x,q) - V_0(y,q) - \frac{|x-y|^2}{\epsilon^2}.$$

We can assume that Ψ_ϵ attains its maximum in $\mathbb{R}^d \times \mathbb{R}^d$ at a point (x_0, y_0):

$$\Psi_\epsilon(x_0,y_0) \geq \Psi_\epsilon(x,y) \quad \forall (x,y) \in \mathbb{R}^d \times \mathbb{R}^d \tag{4.3.13}$$

First, we note that

$$\Psi_\epsilon(x_0,x_0) \leq \Psi_\epsilon(x_0,y_0)$$

hence

$$V_{h0}(x_0,q) - V_0(x_0,q) \leq V_{h0}(x_0,q) - V_0(y_0,q) - \frac{|x_0-y_0|^2}{\epsilon^2}.$$

From the Lipschitz continuity of V_0 we derive

$$\frac{|x_0-y_0|^2}{\epsilon^2} \leq V_0(x_0,q) - V_0(y_0,q) \leq L_{V_0}|x_0-y_0|$$

and obtain

$$|x_0-y_0| \leq L_{V_0}\epsilon^2. \tag{4.3.14}$$

Now we define

$$\phi_\epsilon(y) := V_{h0}(x_0,q) - \frac{|x_0-y|^2}{\epsilon^2}$$

and from (4.3.13) we have that the function $y \mapsto V_0(y,q) - \phi_\epsilon(y)$ attains its minimum at y_0. Since V_0 is a viscosity solution of (4.1.1), there exists $\tilde{u} \in \mathcal{U}$ such that

$$\lambda V_0(y_0,q) - \ell(y_0,q,\tilde{u}) - f(y_0,q,\tilde{u}) \cdot D\phi_\epsilon(y_0,q) \geq 0$$

or, more explicitly

$$\lambda V_0(y_0,q) \geq \ell(y_0,q,\tilde{u}) + f(y_0,q,\tilde{u}) \cdot \frac{2|x_0-y_0|}{\epsilon^2}. \tag{4.3.15}$$

On the other hand, combining the fact that V_{h0} is solution to (4.1.14) with assumption (S3) we have

$$\lambda V_{h0}(x_0,q) + H\big(x_0,q,DV_{h0}(x_0,q)\big) \leq K_c\mathcal{E}(h,V_{h0})$$

which implies

$$\lambda V_{h0}(x_0,q) \leq \ell(x_0,q,\tilde{u}) + f(x_0,q,\tilde{u}) \cdot \frac{2|x_0-y_0|}{\epsilon^2} + K_c\mathcal{E}(h,V_{h0}). \tag{4.3.16}$$

By subtracting (4.3.15) from (4.3.16) we obtain

$$\begin{aligned}\lambda V_{h0}(x_0,q) - \lambda V_0(y_0,q) \leq \\ \leq \ell(x_0,q,\tilde{u}) - \ell(y_0,q,\tilde{u}) + \\ + (f(x_0,q,\tilde{u}) - f(y_0,q,\tilde{u})) \cdot \frac{2|x_0-y_0|}{\epsilon^2} + \\ + K_c\mathcal{E}(h,V_{h0})\end{aligned}$$

which, because of the Lipschitz continuity of ℓ and f, leads to

$$\begin{aligned}V_{h0}(x_0,q) - V_0(y_0,q) \leq \frac{L_\ell}{\lambda}|x_0-y_0| + \frac{2|x_0-y_0|}{\epsilon^2}\frac{L_f}{\lambda}|x_0-y_0| + \\ + \frac{K_c}{\lambda}\mathcal{E}(h,V_{h0}).\end{aligned}$$

From (4.3.13) it follows

$$V_{h0}(x,q) - V_0(x,q) \leq V_{h0}(x_0,q) - V_0(y_0,q) - \frac{|x_0-y_0|^2}{\epsilon^2}. \tag{4.3.17}$$

hence

$$\begin{aligned}V_{h0}(x,q) - V_0(x,q) \leq \frac{|x_0-y_0|^2}{\epsilon^2} + \frac{L_\ell}{\lambda}|x_0-y_0| + \\ + \frac{2|x_0-y_0|}{\epsilon^2}\frac{L_f}{\lambda}|x_0-y_0| + \frac{K_c}{\lambda}\mathcal{E}(h,V_{h0}).\end{aligned}$$

By using (4.3.14) on the previous estimate, we have

$$V_{h0}(x,q) - V_0(x,q) \leq \left(L_{V_0} + \frac{L_\ell}{\lambda} + \frac{L_f}{\lambda}2L_{V_0}\right)L_{V_0}\epsilon^2 + \frac{K_c}{\lambda}\mathcal{E}(h,V_{h0})$$

and if we choose

$$\epsilon = \sqrt{\frac{K_c\mathcal{E}(h,V_{h0})}{(\lambda L_{V_0} + L_\ell + L_f 2L_{V_0})L_{V_0}}}$$

we obtain,

$$V_{h0}(x,q) - V_0(x,q) \leq 2\frac{K_c}{\lambda}\mathcal{E}(h,V_{h0}) \leq 2\frac{K_c}{\lambda}\mathcal{E}(h,V_{h0})$$

The same estimate on the difference $V_0(x,q) - V_{h0}(x,q)$ can be proved by applying the previous steps to the test function

$$V_0(x,q) - V_{h0}(y,q) - \frac{|x-y|^2}{\epsilon^2}.$$

Now, by using the definition of γ in (4.3.12) we obtain

$$\mathcal{E}(h,V_{h0}) := \sum_{i\in J}|D^iV_{h0}|_0|h|^{k_i} \leq |J|L_{V_{h0}}|V_{h0}|_0|h|^\gamma$$

and, since we assumed $\lambda > 1$, by defining

$$C_0 := 2K_c|J| \max\{L_{V_0}|V_0|_0, L_{V_{h0}}|V_{h0}|_0\}$$

we conclude

$$\left|V_{h0}(x,q) - V_0(x,q)\right| \leq C_0|h|^\gamma. \tag{4.3.18}$$

□

4.3.3 The error estimate for the problem with n switches

First, for every $0 < \varepsilon < \frac{\beta}{4}$ (where β is as in (A2)), we define V_n^ε as the viscosity solution of

$$\begin{aligned} \max\Big\{&\lambda V_n(x,q) + \max_{|e|\leq\varepsilon} H(x+e,q,D_xV_n(x,q)), \\ &V_n(x,q) - \mathcal{N}V_{n-1}(x,q)\Big\} = 0 \end{aligned} \quad \text{on } \mathbb{R}^d \times \mathbb{I}. \tag{4.3.19}$$

We recall that the fact that (4.3.19) has a unique solution as a consequence of Lemma 4.3.1 (i).

Lemma 4.3.3. *Let V_n^ε be the viscosity solution of (4.3.19), for $n \geq 1$. Then, an upper bound of the Lipschitz constant of V_n^ε is*

$$|V_n^\varepsilon|_1 \leq \max\{L_{V_0}, L_{c_\mathcal{C}}\}. \tag{4.3.20}$$

Proof. Using the same methods as for sequence (4.1.3), we can show that

$$0 \leq \cdots \leq V_n^\varepsilon \leq \cdots \leq V_2^\varepsilon \leq V_1^\varepsilon \leq V_0^\varepsilon \tag{4.3.21}$$

Combining with (4.3.20), get

$$0 \leq \cdots \leq L_{V_n^\varepsilon} \leq \cdots \leq L_{V_2^\varepsilon} \leq L_{V_1^\varepsilon} \leq \max\{L_{V_0}, L_{c_\mathcal{C}}\} \tag{4.3.22}$$

□

We can give now the error estimate of the upper and lower bound of the difference between V_n and V_{hn}. We recall that C_0 has been defined in Proposition 4.3.2.

Proposition 4.3.3. *For $n \geq 1$, let $V_n \in C_{b,l}(\mathbb{R}^d \times \mathbb{I})$ be the unique viscosity solution of (4.1.2), and $V_{hn} \in C_b(\mathbb{R}^d \times \mathbb{I})$ the unique solution of (4.1.16). Then, on $\mathbb{R}^d \times \mathbb{I}$ we have*

$$-\underline{C}_n|h|^\gamma \leq V_n(x,q) - V_{hn}(x,q) \leq \overline{C}_n|h|^\gamma \tag{4.3.23}$$

where, for every $n \geq 1$, there exist positive constants $\overline{K}_{V_{n-1}}$ and $\underline{K}_{V_{h(n-1)},h}$ such that

$$\begin{aligned} \overline{C}_n &:= \overline{C}_{n-1} + \overline{K}_{V_{n-1}} \\ \underline{C}_n &:= \underline{C}_{n-1} + \underline{K}_{V_{h(n-1)},h}. \end{aligned} \tag{4.3.24}$$

Proof. . We prove the proposition by induction over n, starting from the upper bound.

Let $n = 1$. We want to estimate the difference

$$V_1(x,q) - V_{h1}(x,q) = V_1(x,q) - \tilde{V}_{h1}(x,q) + \tilde{V}_{h1}(x,q) - V_{h1}(x,q)$$

where $\tilde{V}_{h1}$ is the solution of

$$\max\left\{S(x,q,\tilde{V}_{h1}(x,q),\tilde{V}_{h1}), \tilde{V}_{h1}(x,q) - \mathcal{N}V_0(x,q)\right\} = 0 \quad \text{on } \mathbb{R}^d \times \mathbb{I}.$$

By applying Proposition 4.3.1, (4.3.11) and Proposition 4.3.2 we obtain

$$V_1(x,q) - V_{h1}(x,q) \leq \overline{K}_{V_1,\mathcal{N}V_0}|h|^\gamma + C_0|h|^\gamma.$$

Note that, for every $n \geq 1$, the constant $\overline{K}_{V_n,\mathcal{N}V_{n-1}}$ coincides with the constant $\overline{K}_{w,\Phi}$ defined in Proposition 4.3.1 in the case $w = V_n$ and $\Phi = \mathcal{N}V_{n-1}$ and, by assumption (A4), it can be simplified to

$$\begin{aligned}
\overline{K}_{V_{n-1}} &:= \frac{L_\ell}{\lambda - L_f} + L_{V_{n-1}}\frac{L_f}{\lambda} + L_{V_{n-1}} + K_c L_{V_{n-1}}|V_{n-1}|_0|J| \geq \\
&\geq \frac{L_\ell}{\lambda - L_f} + L_{V_{n-1}}\frac{L_f}{\lambda} + L_{V_n} + K_c L_{V_n}|V_n|_0|J| \geq \\
&\geq \frac{L_\ell}{\lambda - L_f} + L_{\mathcal{N}V_{n-1}}\frac{L_f}{\lambda} + L_{V_n} + K_c L_{V_n}|V_n|_0|J| =: \overline{K}_{V_n,\mathcal{N}V_{n-1}}.
\end{aligned} \tag{4.3.25}$$

We also recall that

$$C_0 := 2K_c|J|\max\{L_{V_0}|V_0|_0, L_{V_{h0}}|V_{h0}|_0\}$$

while K_c and J are defined in the consistency hypothesis (S3).

By the definition of $\overline{K}_{V_0}$ in (4.3.25), we obtain

$$V_1(x,q) - V_{h1}(x,q) \leq \overline{K}_{V_0}|h|^\gamma + C_0|h|^\gamma$$

and, defining $\overline{C}_1 := \overline{K}_{V_0} + C_0$, we have

$$V_1(x,q) - V_{h1}(x,q) \leq \overline{C}_1|h|^\gamma \quad \forall (x,q) \in \mathbb{R}^d \times \mathbb{I}.$$

Let us now suppose that the result is true for n. For $n+1$, applying Proposition 4.3.1 and (4.3.11) we have

$$\begin{aligned}
V_{n+1}(x,q) - V_{h(n+1)}(x,q) &= V_{n+1}(x,q) - \tilde{V}_{h(n+1)}(x,q) + \\
&\quad + \tilde{V}_{h(n+1)}(x,q) - V_{h(n+1)}(x,q) \leq \\
&\leq \overline{K}_{V_n}|h|^\gamma + \left|V_n(x,q) - V_{hn}(x,q)\right| \leq \\
&\leq \overline{K}_{V_n}|h|^\gamma + \overline{C}_n|h|^\gamma.
\end{aligned}$$

Hence, by taking $\overline{C}_{n+1} := \overline{K}_{V_n} + \overline{C}_n$ we finally obtain

$$V_{n+1}(x,q) - V_{h(n+1)}(x,q) \leq \overline{C}_{n+1}|h|^\gamma.$$

For the lower bound, the base case of the induction can be obtained in a similar way by applying Proposition 4.3.1 and (4.3.11):

$$V_{h1}(x,q) - V_1(x,q) = V_{h1}(x,q) - \tilde{V}_{h1}(x,q) + \tilde{V}_{h1}(x,q) - V_1(x,q) \leq \\ \leq C_0|h|^\gamma + \underline{K}_{V_{h0},h}|h|^\gamma$$

and then defining $\underline{C}_1 := \underline{K}_{V_{h0},h} + C_0$, where, for every $n \geq 1$, the constant $\underline{K}_{V_{hn},h}$ coincides with $\underline{K}_{w_h,h}$ defined in Proposition 4.3.1 in the case $w_h = V_{hn}$:

$$\underline{K}_{V_{hn},h} := \max\left\{(L_\ell + L_{V_{hn}}L_f)h, \frac{L_\ell + L_{V_{hn}}L_f}{\lambda}\right\} + L_{V_{hn}} + K_c L_{V_{hn}}|V_{hn}|_0|J|.$$

The rest of the induction follows the same steps of the previous case, leading to

$$V_{h(n+1)}(x,q) - V_{n+1}(x,q) \leq \underline{C}_{n+1}|h|^\gamma$$

with $\underline{C}_{n+1} := \underline{K}_{V_{hn},h} + \underline{C}_n$. □

We now set

$$\overline{D}_{n-1} := \overline{C}_n - \overline{C}_{n-1} = \overline{K}_{V_{n-1}}.$$

where $\overline{C}_n$ has been defined in (4.3.24). The definition of K_Φ in Lemma 4.3.1 (ii) and (4.3.22) imply that $\overline{D}_n \leq \overline{D}_0$, and hence

$$\overline{C}_n \leq C_0 + n\overline{D}_0. \tag{4.3.26}$$

Similarly, if we define

$$\underline{D}_{n-1} := \underline{C}_n - \underline{C}_{n-1} = \underline{K}_{V_{h(n-1)},h}$$

from the definition of $K_{w_h,h}$ in (4.3.10) and (4.1.20) we have that $\underline{D}_n \leq \underline{D}_0$, and hence:

$$\underline{C}_n \leq C_0 + n\underline{D}_0 \tag{4.3.27}$$

Before stating our main result, it is important to point out that we used the cascade technique as a tool for obtaining some theoretical error estimates, and not for actually solving the hybrid control problem numerically. We refer to Chapter 3 for a detailed construction of a numerical scheme capable of computing the approximated solution.

Theorem 4.3.1. *Assume (A1)-(A8) and (S1)-(S4). Let $V \in C_{b,l}(\mathbb{R}^d)$ be the unique viscosity solution of (2.2.13), and $V_h \in C_b(\mathbb{R}^d)$ the unique solution of (4.1.14). Then there exist $\overline{C} > 0$ and $\underline{C} > 0$ such that*

$$-\underline{C}\ln h|h|^\gamma \leq V(x,q) - V_h(x,q) \leq \overline{C}\ln h|h|^\gamma \quad \forall (x,q) \in \mathbb{R}^d \times \mathbb{I} \tag{4.3.28}$$

Proof. We start with the upper bound. By (4.1.5), (4.3.26) and (4.1.19) we obtain the following estimate

$$\begin{aligned} V - V_h &= V - V_n + V_n - V_{hn} + V_{hn} - V_h \leq \\ &\leq \frac{(1-\mu)^n}{\mu}|V_0|_0 + (C_0 + n\overline{D}_0)|h|^\gamma + \frac{(1-\mu)^n}{\mu}|V_{h0}|_0 \end{aligned}$$

which can be rearranged as

$$V - V_h \leq \frac{|V_0|_0 + |V_{h0}|_0}{\mu}(1-\mu)^n + \overline{D}_0|h|^\gamma n + C_0|h|^\gamma.$$

For the lower bound, by following the same reasoning and using (4.3.27) instead of (4.3.26) we have

$$V_h - V \leq \frac{(1-\mu)^n}{\mu}|V_{h0}|_0 + (C_0 + n\underline{D}_0)|h|^\gamma + \frac{(1-\mu)^n}{\mu}|V_0|_0$$

or, equivalently

$$V_h - V \leq \frac{|V_0|_0 + |V_{h0}|_0}{\mu}(1-\mu)^n + \underline{D}_0|h|^\gamma n + C_0|h|^\gamma.$$

The idea now is to minimize with respect to n the estimates on the upper and lower bound:

$$\begin{aligned} \overline{E}(n) &:= a(1-\mu)^n + \overline{b}n + c \\ \underline{E}(n) &:= a(1-\mu)^n + \underline{b}n + c \end{aligned}$$

where $a := \frac{|V_0|_0+|V_{h0}|_0}{\mu}$, $\overline{b} := \overline{D}_0|h|^\gamma$, $\underline{b} := \underline{D}_0|h|^\gamma$ and $c := C_0|h|^\gamma$.

By a straightforward application of [12, Lemma 6.1] to $\overline{E}(n)$ we have

$$\begin{cases} V - V_h \leq -\frac{\overline{b}}{\ln(1-\mu)} + c & -\frac{\overline{b}}{a\ln(1-\mu)} \geq 1 \\ V - V_h \leq -\frac{(1-\mu)\overline{b}}{\ln(1-\mu)} + \overline{b}\left(\log_{1-\mu}\left(-\frac{\overline{b}}{a\ln(1-\mu)}\right) + 1\right) + c & \text{else.} \end{cases}$$

More explicitly, if

$$-\frac{\overline{D}_0|h|^\gamma}{\frac{|V_0|_0+|V_{h0}|_0}{\mu}\ln(1-\mu)} \geq 1$$

then the upper bound for $V - V_h$ is

$$\left(-\frac{\overline{D}_0}{\ln(1-\mu)} + C_0\right)|h|^\gamma$$

otherwise

$$\left(-\frac{(1-\mu)\overline{D}_0}{\ln(1-\mu)} + \overline{D}_0\left(\log_{1-\mu}\left(-\frac{\mu\overline{D}_0|h|^\gamma}{(|V_0|_0 + |V_{h0}|_0)\ln(1-\mu)}\right) + 1\right) + C_0\right)|h|^\gamma$$

in this second case, the factor multiplying $|h|^\gamma$ is $O(\ln h) + O(1)$.

The same can be proven for the lower bound, replacing $\overline{D}_0$ with $\underline{D}_0$, thus proving the result. □

Appendix A

Appendix to Chapter 4

A.1 The upper bounds of the Lipschitz constants

Proof of Lemma 4.2.2. Set

$$m_\epsilon := \sup_{x,y} \varphi(x,y) := \sup_{x,y\in\mathbb{R}^d} \{w(x,q) - w(y,q) - \frac{\delta}{2}|x-y|^2 - \frac{\epsilon}{2}(|x|^2+|y|^2)\}.$$

Let $x_0, y_0 \in \mathbb{R}^d$ such that $m_\epsilon = \varphi(x_0, y_0)$. Taking into account the HJB equation satisfied by w and applying the viscosity notion, we get:

$$\begin{aligned} 0 \leq \max\{&\lambda w(y_0,q) + H(y_0,q,p_y) - \lambda w(x_0,q) - H(x_0,q,p_x),\\ &w(y_0,q) - \Phi(y_0,q) - w(x_0,q) + \Phi(x_0,q)\} \end{aligned}$$

where

$$\begin{aligned} p_x &= \delta(x_0 - y_0) + \epsilon x_0 \\ p_y &= \delta(x_0 - y_0) - \epsilon y_0. \end{aligned} \tag{A.1.1}$$

Two cases have to be considered.

a) $\lambda w(y_0,q) + H(y_0,q,p_y) - \lambda w(x_0,q) - H(x_0,q,p_x)$.

This is the standard case (see [3]), and we have that

$$w(x,q) - w(y,q) \leq \frac{L_\ell}{\lambda - L_f}|x-y| \quad \forall x,y \in \mathbb{R}^d.$$

b) $w(y_0,q) - \Phi(y_0) - w(x_0,q) + \Phi(x_0)$.

In this case, we get $w(x_0,q) - w(y_0,q) \leq L_\Phi|x_0 - y_0|$. Then we deduce that

$$m_\epsilon \leq L_\Phi|x_0 - y_0| - \frac{\delta}{2}|x_0 - y_0|^2. \tag{A.1.2}$$

Setting $r := |x_0 - y_0|$, and noting that $\max_{r\geq 0}\left\{L_\Phi r - \frac{\delta}{2}r^2\right\} = L_\Phi^2/2\delta$, we obtain

$$m_\epsilon \leq \frac{L_\Phi^2}{2\delta}.$$

Applying a simple calculus argument (see [37, Lemma 2.3]), for fixed δ, we have:

$$\lim_{\epsilon \to 0} m_\epsilon = \sup_{x,y \in \mathbb{R}^d} \{w(x,q) - w(y,q) - \delta |x-y|^2\} := m$$

and hence $m \leq \frac{L_\Phi^2}{2\delta}$.

Therefore, by definition of m, we have that:

$$w(x,q) - w(y,q) \leq \frac{L_\Phi^2}{2\delta} + \frac{\delta}{2}|x-y|^2 \quad \forall x, y \in \mathbb{R}^d.$$

Now by using the simple remark that

$$\min_{\delta \geq 0} \left\{ \frac{L_\Phi^2}{2\delta} + \frac{\delta}{2}|x-y|^2 \right\} = L_\Phi |x-y|$$

we obtain:

$$w(x,q) - w(y,q) \leq L_\Phi |x-y| \quad \forall x, y \in \mathbb{R}^d.$$

In conclusion, for the two above cases, we obtain

$$L_w = \max \left\{ L_\Phi, \frac{L_\ell}{\lambda - L_f} \right\}$$

By using similar arguments, we can compute L_{w^ϵ} and get:

$$L_{w^\epsilon} = \max \left\{ L_\Phi, \frac{L_\ell}{\lambda - L_f} \right\}.$$

□

A.2 Lipschitz stability for the SL scheme

In this section we will prove that, in the case described in Remark 4.0.1, the numerical approximation w_h of w of the obstacle problem is Lipschitz continuous.

We consider schemes approximating (4.2.1) in the form:

$$W_h(x,q) = \min \{\Sigma^h(x,q,W_h), \Phi(x,q)\} \quad \text{on } x \in \mathbb{R}^d \times \mathbb{I}. \tag{A.2.1}$$

In the case of a Semi-Lagrangian scheme, we recall the definition (3.3.5) of the operator Σ^h:

$$\Sigma^h(x,q,W_h) := \min_{u \in U} \Big\{ h\ell(x,q,u) + e^{-\lambda h} \mathcal{I}[W_h](x + hf(x,q,u), q) \Big\}.$$

It is well-known that Σ^h is non-expansive in the ∞-norm.

Theorem A.2.1. *Under assumptions (A1)-(A8), (S1)-(S4), the solution W_h of problem* (4.3.1) *obtained with the Semi-Lagrangian scheme* (A.2.1) *is Lipschitz continuous with*

$$|W_h|_1 \leq L_{W_h} = (1 + (\lambda - L_f)h) \max\left\{L_\Phi, \frac{L_\ell}{\lambda - L_f}\right\}.$$

Proof. For any $q \in \mathbb{I}$, let us consider the solution of the numerical scheme Σ^h in fixed point form:

$$W_h^{(k+1)}(x,q) = \min\left\{\Sigma^h(x,q,W_h^{(k)}), \Phi(x,q)\right\}$$

where $W_h^{(k)}$ is the approximation of W_h at iteration k.

For any $x_1, x_2 \in \mathbb{R}^d$ we have

$$\begin{aligned}
&|W_h^{(k+1)}(x_1,q) - W_h^{(k+1)}(x_2,q)| \leq \\
&\quad \leq \max\left\{hL_\ell + e^{-\lambda h}(1+\lambda L_f)L_{W_h^{(k)}}, L_\Phi\right\} \leq \\
&\quad \leq \max\{hL_\ell + e^{-(\lambda - L_f)h}L_{W_h^{(k)}}, L_\Phi\} \leq \\
&\quad \leq \max\left\{\frac{L_\ell}{\lambda - L_f}, L_\Phi\right\} \max\left\{(\lambda - L_f)h + \frac{e^{-(\lambda-L_f)h}L_{W_h^{(k)}}}{\max\left\{\frac{L_\ell}{\lambda-L_f}, L_\Phi\right\}}, 1\right\}.
\end{aligned}$$

By setting

$$m := (\lambda - L_f)h$$
$$M_k := \frac{L_{W_h^{(k)}}}{\max\left\{\frac{L_\ell}{\lambda-L_f}, L_\Phi\right\}}$$

we have

$$M_{k+1} \leq \max\{m + e^{-m}M_k, 1\}.$$

Note that if $M_k \leq 1 + m$, then $M_{k+1} \leq 1 + m$ because the inequality $M_k \leq 1 + m$ implies $e^{-m}M_k \leq e^{-m}(1 + m) \leq 1$. It follows

$$M_{k+1} \leq \max\{m + e^{-m}M_k, 1\} \leq \max\{1 + m, 1\} = 1 + m.$$

It suffices then to initialize the fixed point iterations with $W_h^{(0)}$ such that $M_0 = 0$ to guarantee $M_k \leq 1 + m$ for every $k \geq 0$, and, by the definitions of m and M_k, we obtain

$$\frac{L_{W_h^{(k)}}}{\max\left\{\frac{L_\ell}{\lambda-L_f}, L_\Phi\right\}} \leq 1 + (\lambda - L_f)h$$

which implies

$$L_{W_h^{(k)}} \leq (1 + (\lambda - L_f)h) \max\left\{\frac{L_\ell}{\lambda - L_f}, L_\Phi\right\}.$$

Now, since $W_h^{(k)}$ converges towards the solution W_h of the scheme (A.2.1) as $k \to +\infty$, we conclude

$$L_{W_h} \leq (1 + (\lambda - L_f)h) \max\left\{\frac{L_\ell}{\lambda - L_f}, L_\Phi\right\}.$$

□

A.3 Estimate on the perturbed value function of the stopping problem

For the sensitivity analysis in this section, we will drop the assumptions $\mathcal{C} = \mathbb{R}^d$ and $\mathcal{A} = \emptyset$ made in 4.0.1 and consider the more general case in which the two sets just non-empty.

Let us consider the controlled system defined in (2.2.1) with a variation: the ability to switch between dynamics is replaced with the ability to stop when the trajectory enters the two predefined sets $\mathcal{A}$ and $\mathcal{C}$. More precisely:

- On hitting $\mathcal{A}$ the trajectory has to stop.
- When the trajectory evolves in the set $\mathcal{C}$, the controller can choose to stop or not. In this case the stopping time is denoted by ξ and, if controller doesn't choose to stop, its value is $+\infty$ by definition.
- If the trajectory is neither inside $\mathcal{A}$ or $\mathcal{C}$, it cannot stop.

The control strategy θ consists then in the continuous control u and the controlled stopping time ξ (which can be finite or infinite) and belongs to the set $\Theta := \mathcal{U} \times \mathbb{R}_+$. Throughout this section we will assume all the basic hypotheses (A1)-(A7).

In order to define the cost associated to a control strategy (u, ξ) we need to define the notion of *hitting time* for this problem.

Definition A.3.1 (Hitting time)**.** *Let $E \subset \mathbb{R}^d$, $x \in \mathbb{R}^d \setminus E$, $\theta \in \Theta$ and $q \in \mathbb{I}$. We define the hitting time of the trajectory X associated to the set E as*

$$t_E(x, q, \theta) := \inf\{t > 0 \quad | \quad X_x(t, q, u) \in E\}$$

if the trajectory never enters the set E we set $t_E = +\infty$ by definition. Note that this definition implies $\xi > t_{\mathcal{C}_q}(x, q, \theta)$.

The cost can now be defined as

$$\begin{aligned} J(x, q; \theta) := &\int_0^\xi \ell(X_x(t, q, u), q, u(t)) e^{-\lambda t} dt + \\ &+ e^{-\lambda \xi} \Phi(X_x(\xi, q, u), q) + \\ &+ e^{-\lambda t_{\mathcal{A}_q}(x,q,\theta)} \Psi\Big(X_x(t_{\mathcal{A}_q}(x, q, \theta), q, u), q\Big) \end{aligned}$$

where $\Phi : \mathcal{C} \to \mathbb{R}$ and $\Psi : \mathcal{A} \to \mathbb{R}$ are Lipschitz continuous, and the value function of this problem

$$w(x,q) := \inf_{\theta \in \Theta} J(x,q;\theta)$$

satisfies (4.2.1).

For $\varepsilon > 0$ small enough and for every $q \in \mathbb{I}$, we consider the set $\mathcal{A}_q^\varepsilon$ defined by:

$$\mathcal{A}_q^\varepsilon := \{x \in \mathcal{A}_q^\varepsilon \quad | \quad d_{\mathcal{A}_q^\varepsilon}(x) \leq -\varepsilon\}$$

where, for a set $E \in \mathbb{R}^d$, $d_E(x)$ is the signed distance between the point x and ∂E:

$$d_E(x) := \begin{cases} d(x,\partial E) & x \in \bar{E}^{\mathrm{c}} \\ 0 & x \in \partial E \\ -d(x,\partial E) & x \in \mathring{E} \end{cases}$$

Note that by assumption (A1), the boundary of $\mathcal{A}_q^\varepsilon$ is C^2. We define also $\mathcal{A}^\varepsilon := \bigcup \mathcal{A}_q^\varepsilon \times \{q\}$.

If we replace the dynamics in (2.2.1) with

$$\begin{cases} \dot{X}^\varepsilon(t) = f(X^\varepsilon(t) + e(t), Q(t), u(t)) \\ X^\varepsilon(0) = x \\ Q(0^+) = q \end{cases}$$

where, given $\varepsilon > 0$, $e \in \mathcal{F}^\varepsilon$ with

$$\mathcal{F}^\varepsilon := \{e : (0,+\infty) \to \mathbb{R}^d \quad | \quad e \text{ measurable, } |e(t)| \leq \varepsilon \text{ a.e.}\}$$

we can define in a similar way the value function w^ε, solution to the system

$$\begin{cases} \lambda w^\varepsilon(x,q) + \max\limits_{|e| \leq \varepsilon} H\big(x+e, q, D_x w^\varepsilon(x,q)\big) = 0 & \text{on } \mathcal{O}^\varepsilon \qquad \text{(A.3.1a)} \\ \max\Big\{\lambda w^\varepsilon(x,q) + \max\limits_{|e| \leq \varepsilon} H\big(x+e, q, D_x w^\varepsilon(x,q)\big), \\ \qquad w^\varepsilon(x,q) - \Phi(x,q)\Big\} = 0 & \text{on } \mathcal{C} \qquad \text{(A.3.1b)} \\ w^\varepsilon(x,q) - \Psi(x,q) = 0 & \text{on } \mathcal{A}^\varepsilon \qquad \text{(A.3.1c)} \end{cases}$$

where $\mathcal{O}^\varepsilon := \mathbb{R}^d \times \mathbb{I} \setminus (\mathcal{A}^\varepsilon \cup \mathcal{C})$ and

$$w^\varepsilon(x,q) := \inf_{\theta^\varepsilon \in \Theta^\varepsilon} J^\varepsilon(x,q;\theta^\varepsilon)$$

with $\Theta^\varepsilon := \mathcal{U} \times \mathbb{R}_+ \times \mathcal{F}^\varepsilon$, and, for a sufficiently small ε

$$\begin{aligned} J^\varepsilon(x,q;\theta^\varepsilon) := \int_0^{\xi^\varepsilon} & \ell\big(X_x^\varepsilon(t,q,u) + e(t), q, u(t)\big) e^{-\lambda t} dt + \\ & + e^{-\lambda \xi^\varepsilon} \Phi\big(X_x^\varepsilon(\xi^\varepsilon, q, u), q\big) + \\ & + e^{-\lambda t^\varepsilon_{\mathcal{A}_q^\varepsilon}(x,q,\theta)} \Psi\Big(X_x^\varepsilon\big(t^\varepsilon_{\mathcal{A}_q^\varepsilon}(x,q,\theta), q, u\big), q\Big) \end{aligned}$$

We use the notation $\cdot^\varepsilon$ to distinguish between the quantities related to the perturbed trajectory and the ones related to the unperturbed trajectory: ξ^ε is the stopping time of X_x^ε inside the set $\mathcal{C}_q^\varepsilon$, and t_E^ε is the hitting time of the perturbed trajectory relative to the set E.

The rest of this section is dedicated to the results necessary for obtaining an estimate for the difference between w^ε and w, respectively solutions of (4.3.5) and (4.2.1).

A.3.1 Estimate for the hitting times

In order to prove that the magnitude of the hitting times of the perturbed and unperturbed trajectory can be controlled by means of ε, we first need an estimate on $t_{\mathcal{A}_q}(x,q,\theta)$ and $t^\varepsilon_{\mathcal{A}_q^\varepsilon}(x,q,\theta)$ for points close to $\mathcal{A}_q$ and $\mathcal{A}_q^\varepsilon$ respectively.

Lemma A.3.1. *Let $\theta \in \Theta$, $q \in \mathbb{I}$ and $\omega > 0$ as defined in (A5). Then the following statements are true:*

i) There exists $\delta > 0$ such that

$$t_{\mathcal{A}_q}(x,q,\theta) < \frac{d_{\mathcal{A}_q}(x)}{\omega} \quad \forall x \in B(\partial\mathcal{A}_q,\delta) \setminus \mathcal{A}_q$$

ii) There exist $\delta > 0$ and $\bar\varepsilon > 0$ such that

$$t^\varepsilon_{\mathcal{A}_q^\varepsilon}(x,q,\theta) < \frac{2d_{\mathcal{A}_q^\varepsilon}(x)}{\omega} \quad \forall x \in B(\partial\mathcal{A}_q^\varepsilon,\delta) \setminus \mathcal{A}_q^\varepsilon, \forall \varepsilon < \bar\varepsilon$$

The same results are also true replacing the set $\mathcal{A}_q$ with the set $\mathcal{C}_q$.

Proof. We will prove the first statement, then use an analogous argument to prove the second. Once the two statements are proven to be true for the set $\mathcal{A}_q$, the fact that they hold for the set $\mathcal{C}_q$ is trivial.

i) Since $\mathcal{A}_q$ has a C^2 boundary, we can choose $\bar r > 0$ such that $d_{\mathcal{A}_q}$ is C^1 on $B(\partial\mathcal{A}_q,\bar r)$. Then we take $r < \bar r$ such that

$$f(x,q,u)\cdot d'_{\mathcal{A}_q}(x) < -\omega \quad \forall x \in B(\partial\mathcal{A}_q,r)$$

Let $x \in B(\partial\mathcal{A}_q,r)$, choosing $\bar t > 0$ such that $X_x(t,q,u)$ is in $B(\partial\mathcal{A}_q,r)$ for every $t < \bar t$, we have

$$d_{\mathcal{A}_q}(X_x(\bar t,q,u)) - d_{\mathcal{A}_q}(x) = \int_0^{\bar t} d'_{\mathcal{A}_q}(X_x(t,q,u))f(X_x,q,u)dt < -\omega\bar t$$

On the other hand, taking $\delta = \min\{r,\omega\bar t\}$ we have

$$x \in B(\partial\mathcal{A}_q,\delta) \Rightarrow d_{\mathcal{A}_q}(x) < \omega\bar t$$

Now, for $t^\delta = \frac{d(x)}{\omega}$ we get

$$d_{\mathcal{A}_q}(X_x(t^\delta, q, u)) < -\omega t^\delta + d_{\mathcal{A}_q}(x) = 0$$

From the definition of $d_{\mathcal{A}_q}$, this implies that $X_x(t^\delta, q, u) \in \mathring{\mathcal{A}}_q$ and since $t_{\mathcal{A}_q} < t^\delta$

$$t_{\mathcal{A}_q}(x, q, \theta) < \frac{d_{\mathcal{A}_q}(x)}{\omega} \quad \forall x \in B(\partial\mathcal{A}_q, \delta) \setminus \mathcal{A}_q$$

ii) If we choose $r > 0$ as in the previous step of the proof, from the definition of $\mathcal{A}_q^\varepsilon$ we have that

$$B\left(\partial\mathcal{A}_q^\varepsilon, \frac{r}{2}\right) \subset B\left(\partial\mathcal{A}_q, r\right) \quad \forall \varepsilon \leq \frac{r}{2}$$

and

$$d_{\mathcal{A}_q^\varepsilon}(x) = d_{\mathcal{A}_q}(x) + \varepsilon \quad \forall x \in \mathbb{R}^d, \forall \varepsilon > 0$$

taking $\varepsilon \leq \frac{r}{2}$, x in $B\left(\partial\mathcal{A}_q^\varepsilon, \frac{r}{2}\right)$ and choosing $\bar{t} > 0$ such that the trajectory $X_x^\varepsilon(t, q, u)$ is in $B\left(\partial\mathcal{A}_q^\varepsilon, \frac{r}{2}\right)$ for every $t < \bar{t}$, we have

$$\begin{aligned} d_{\mathcal{A}_q^\varepsilon}(X_x^\varepsilon(\bar{t}, q, u)) - d_{\mathcal{A}_q^\varepsilon}(x) &= \int_0^{\bar{t}} d'_{\mathcal{A}_q}(X_x^\varepsilon(t, q, u)) f(X_x^\varepsilon + e, q, u) dt \leq \\ &\leq \int_0^{\bar{t}} d'_{\mathcal{A}_q}(X_x^\varepsilon(t, q, u)) f(X_x^\varepsilon, q, u) dt + \\ &\quad + \varepsilon M_{d'_{\mathcal{A}_q}} L_f \bar{t} \leq \\ &< -\omega\bar{t} + \varepsilon M_{d'_{\mathcal{A}_q}} L_f \bar{t} \end{aligned}$$

where $M_{d'_{\mathcal{A}_q}}$ is the upper bound of $d'_{\mathcal{A}_q}$ on $B(\partial\mathcal{A}_q^{\frac{r}{2}}, \frac{r}{2})$. If we then choose $\bar{\varepsilon}$ such that

$$\bar{\varepsilon} = \min\left\{\frac{r}{2}, \frac{\omega}{2M_{d'_{\mathcal{A}_q}} L_f}\right\}$$

for $\varepsilon \leq \bar{\varepsilon}$ we have

$$d_{\mathcal{A}_q^\varepsilon}(X_x^\varepsilon(\bar{t}, q, u)) - d_{\mathcal{A}_q^\varepsilon}(x) < -\frac{\omega\bar{t}}{2}$$

hence taking $\delta = \min\left\{\frac{r}{2}, \frac{\omega\bar{t}}{2}\right\}$ we obtain

$$x \in B(\partial\mathcal{A}_q^\varepsilon, \delta) \Rightarrow d_{\mathcal{A}_q^\varepsilon}(x) < \frac{\omega\bar{t}}{2}$$

Now if we take $t^\delta = \frac{2d_{\mathcal{A}_q^\varepsilon}(x)}{\omega}$

$$d_{\mathcal{A}_q^\varepsilon}(X_x^\varepsilon(t^\delta, q, u)) < -\frac{\omega}{2}t^\delta + d_{\mathcal{A}_q^\varepsilon}(x) = 0$$

this again implies that $X_x^\varepsilon(t^\delta, q, u) \in \mathring{\mathcal{A}}_q^\varepsilon$ and since $t^\varepsilon_{\mathcal{A}_q^\varepsilon} < t^\delta$

$$t^\varepsilon_{\mathcal{A}_q^\varepsilon}(x, q, \theta) < \frac{2d_{\mathcal{A}_q^\varepsilon}(x)}{\omega} \quad \forall x \in B(\partial\mathcal{A}_q^\varepsilon, \delta) \setminus \mathcal{A}_q^\varepsilon, \forall \varepsilon < \bar{\varepsilon}$$

□

We will also need the following estimate on the distance between the two trajectories at a given time.

Lemma A.3.2. *Let $x \in \mathbb{R}^d$, $u \in \mathcal{U}$, $q \in \mathbb{I}$ and $\varepsilon > 0$. Then, the difference between the trajectories X^ε and X satisfies the following inequality*

$$\left|X_x^\varepsilon(t, q, u(t)) - X_x(t, q, u(t))\right| \leq \varepsilon(e^{L_f t} - 1) \quad \forall t \geq 0$$

Proof. Since u and q are fixed, we drop the dependency of f on those two variables, for readability. Because f is Lipschitz continuous in the state variable, we have $\forall t \geq 0$

$$\begin{aligned}\dot{X}_x^\varepsilon(t) - \dot{X}_x(t) &= f(X_x^\varepsilon(t) + e(t)) - f(X_x(t)) = \\ &= f(X_x^\varepsilon(t) + e(t)) - f(X_x^\varepsilon(t)) + f(X_x^\varepsilon(t)) - f(X_x(t)) \leq \\ &\leq \varepsilon L_f + L_f|X_x^\varepsilon(t) - X_x(t)|\end{aligned}$$

then applying Grönwall's inequality we obtain

$$|X_x^\varepsilon(t) - X_x(t)| \leq \varepsilon(e^{L_f t} - 1) + |X_x^\varepsilon(0) - X_x(0)|e^{L_f t} = \varepsilon(e^{L_f t} - 1)$$

□

We can now prove the following result.

Lemma A.3.3. *Let $x \in \mathbb{R}^d \setminus (\mathcal{A}_q \cup \mathcal{C}_q)$, $\theta^\varepsilon \in \Theta^\varepsilon$, $\theta \in \Theta$ such that $\theta = \theta^0$ and $q \in \mathbb{I}$, the following statements are true:*

i) If $t_{\mathcal{C}_q}(x, q, \theta) = +\infty$, then $\exists \bar{\varepsilon} > 0$ such that

$$t^\varepsilon_{\mathcal{C}_q}(x, q, \theta^\varepsilon) = +\infty \quad \forall \varepsilon \leq \bar{\varepsilon}$$

The same result is also true when replacing the set $\mathcal{C}_q$ with the set $\mathcal{A}_q$, in this second case we also have $t^\varepsilon_{\mathcal{A}_q^\varepsilon}(x, q, \theta^\varepsilon) = +\infty$.

ii) If $t_{\mathcal{A}_q}(x, q, \theta) < +\infty$, then $\exists \bar{\varepsilon} > 0$ such that $t^\varepsilon_{\mathcal{A}_q^\varepsilon}(x, q, \theta^\varepsilon) < +\infty$ and

$$\left|t^\varepsilon_{\mathcal{A}_q^\varepsilon}(x, q, \theta^\varepsilon) - t_{\mathcal{A}_q}(x, q, \theta)\right| < \varepsilon\frac{2}{\omega}e^{L_f\left(t^\varepsilon_{\mathcal{A}_q^\varepsilon}(x,q,\theta^\varepsilon) \wedge t_{\mathcal{A}_q}(x,q,\theta)\right)} \quad \forall \varepsilon \leq \bar{\varepsilon}$$

where $a \wedge b := \min\{a, b\}$.

iii) If $t_{\mathcal{C}_q}(x,q,\theta) < +\infty$, then $\exists \bar{\varepsilon} > 0$ such that that $t^{\varepsilon}_{\mathcal{C}_q}(x,q,\theta^{\varepsilon}) < +\infty$

Proof. In order to simplify the notation we will again hide the dependency of the hitting times on x, q and u.

i) We will prove the result by contradiction only for $t^{\varepsilon}_{\mathcal{C}_q}$ since the same argument can be used for $t^{\varepsilon}_{\mathcal{A}_q}$. If $t_{\mathcal{C}_q} = +\infty$, Lemma A.3.1 implies that exist $\sigma > 0$ such that

$$d_{\mathcal{C}_q}(yX_x(t,q,u)) > \sigma \quad \forall t \geq 0$$

Suppose that $t^{\varepsilon}_{\mathcal{C}_q} < +\infty$ for every $\varepsilon > 0$, by Lemma A.3.2 we have that

$$\left|X^{\varepsilon}_x(t^{\varepsilon}_{\mathcal{C}_q}) - X_x(t^{\varepsilon}_{\mathcal{C}_q})\right| \leq \varepsilon(e^{L_f t^{\varepsilon}_{\mathcal{C}_q}} - 1) \quad \forall \varepsilon > 0$$

this means that there exists $\varepsilon > 0$ such that $\varepsilon < \frac{\sigma}{e^{L_f t^{\varepsilon}_{\mathcal{C}_q}} - 1}$, but since $X^{\varepsilon}_x(t^{\varepsilon}_{\mathcal{C}_q})$ belongs to $\partial\mathcal{C}_q$ we get

$$X_x(t^{\varepsilon}_{\mathcal{C}_q}) \in B(\partial\mathcal{C}_q, \sigma)$$

which is absurd. Therefore there must exist $\bar{\varepsilon} > 0$ such that

$$t^{\varepsilon}_{\mathcal{C}_q} = +\infty \quad \forall \varepsilon \leq \bar{\varepsilon}$$

Finally, once the result is proven for $t^{\varepsilon}_{\mathcal{A}_q}$, we also have

$$t^{\varepsilon}_{\mathcal{A}_q}(x,q,\theta) \leq t^{\varepsilon}_{\mathcal{A}^{\varepsilon}_q}(x,q,\theta) \Rightarrow t^{\varepsilon}_{\mathcal{A}_q{}^{\varepsilon}}(x,q,\theta) = +\infty$$

ii) For $t < +\infty$ we have

$$\left|X^{\varepsilon}_x(t) - X_x(t)\right| \leq \varepsilon(e^{L_f t} - 1)$$

by Lemma A.3.1 we can take δ and $\bar{\varepsilon}_0$ such that

$$t^{\varepsilon}_{\mathcal{A}^{\varepsilon}_q}(z,q,\theta^{\varepsilon}) < \frac{2d_{\mathcal{A}^{\varepsilon}_q}(z)}{\omega} \quad \forall z \in B(\partial\mathcal{A}^{\varepsilon}_q, \delta) \setminus \mathcal{A}^{\varepsilon}_q, \forall \varepsilon < \bar{\varepsilon}_0$$

note that the definition of $t_{\mathcal{A}_q}$ implies $X_x(t_{\mathcal{A}_q}, q, u) \in \partial\mathcal{A}_q$, hence if we choose $\bar{\varepsilon}$ such that

$$\bar{\varepsilon} = \min\left\{\bar{\varepsilon}_0, \frac{\delta}{2(e^{L_f t_{\mathcal{A}_q}} - 1)}\right\}$$

we have that $X^{\varepsilon}_x(t_{\mathcal{A}_q})$ and $X_x(t^{\varepsilon}_{\mathcal{A}^{\varepsilon}_q})$ belong to $B(\partial\mathcal{A}^{\varepsilon}_q, \delta)$. Therefore, we can have two cases:

- If $t^{\varepsilon}_{\mathcal{A}^{\varepsilon}_q} \leq t_{\mathcal{A}_q}$, then, for every $\varepsilon \leq \bar{\varepsilon}$

$$0 \leq t^{\varepsilon}_{\mathcal{A}^{\varepsilon}_q} - t_{\mathcal{A}_q} = t_{\mathcal{A}_q}(X_x(t^{\varepsilon}_{\mathcal{A}^{\varepsilon}_q}), q, \theta) < \frac{d_{\mathcal{A}_q}(X_x(t^{\varepsilon}_{\mathcal{A}^{\varepsilon}_q}))}{\omega} < \\ < \frac{d_{\mathcal{A}^{\varepsilon}_q}(X_x(t^{\varepsilon}_{\mathcal{A}^{\varepsilon}_q}))}{\omega} \leq \varepsilon \frac{e^{L_f t^{\varepsilon}_{\mathcal{A}_q}} - 1}{\omega}$$

- If $t_{\mathcal{A}_q} \leq t^{\varepsilon}_{\mathcal{A}^{\varepsilon}_q}$, then, for every $\varepsilon \leq \bar{\varepsilon}$

$$0 \leq t_{\mathcal{A}_q} - t^{\varepsilon}_{\mathcal{A}^{\varepsilon}_q} = t^{\varepsilon}_{\mathcal{A}^{\varepsilon}_q}(X^{\varepsilon}_x(t_{\mathcal{A}_q}), q, \theta) < \frac{2 d_{\mathcal{A}^{\varepsilon}_q}(X^{\varepsilon}_x(t_{\mathcal{A}_q}))}{\omega} = \\ = \frac{2 d_{\mathcal{A}_q}(X^{\varepsilon}_x(t_{\mathcal{A}_q})) + 2\varepsilon}{\omega} \leq \varepsilon \frac{2 e^{L_f t_{\mathcal{A}_q}}}{\omega}$$

  Combining the two results, we obtain

$$\left| t^{\varepsilon}_{\mathcal{A}^{\varepsilon}_q}(x, q, \theta^{\varepsilon}) - t_{\mathcal{A}_q}(x, q, \theta) \right| < \varepsilon \frac{2}{\omega} e^{L_f \left(t^{\varepsilon}_{\mathcal{A}^{\varepsilon}_q}(x,q,\theta^{\varepsilon}) \wedge t_{\mathcal{A}_q}(x,q,\theta) \right)} \quad \forall \varepsilon \leq \bar{\varepsilon}$$

- By following the same steps of the previous proof, we have that $t^{\varepsilon}_{\mathcal{C}^{\varepsilon}_q}(x, q, \theta^{\varepsilon})$ is finite, but since the definition of perturbed set implies $t^{\varepsilon}_{\mathcal{C}^{\varepsilon}_q}(x, q, \theta^{\varepsilon}) \geq t^{\varepsilon}_{\mathcal{C}_q}(x, q, \theta^{\varepsilon})$. We also have $t^{\varepsilon}_{\mathcal{C}_q}(x, q, \theta^{\varepsilon}) < +\infty$.

□

A.3.2 Estimate for the cost functionals

The result showed in Lemma A.3.3 proves that, for a sufficiently small $\varepsilon > 0$, the number of different scenarios for the perturbed and unperturbed systems can be reduced to three.

Proposition A.3.1. *Assume (A1)-(A8) and $\lambda > L_f$. There exists $\bar{\varepsilon} > 0$ such that, for every $x \in \mathbb{R}^d \setminus (\mathcal{A}_q \cup \mathcal{C}_q)$, $q \in \mathbb{I}$, $\theta^{\varepsilon} \in \Theta^{\varepsilon}$, $\theta \in \Theta$ such that $\theta = \theta^0$ and $\varepsilon \leq \bar{\varepsilon}$ there can be only three possible behaviors for the trajectories X_x and X^{ε}_x:*

1. *$\xi = +\infty$ and both hitting times $t_{\mathcal{A}_q}(x, q, \theta)$ and $t^{\varepsilon}_{\mathcal{A}^{\varepsilon}_q}(x, q, \theta^{\varepsilon})$ are infinite, neither X_x or X^{ε}_x stop.*

2. *$\xi < +\infty$, X_x and X^{ε}_x enter the sets $\mathcal{C}_q$ and $\mathcal{C}^{\varepsilon}_q$ (respectively) and both stop at time ξ.*

3. *$\xi = +\infty$, X_x and X^{ε}_x enter the sets $\mathcal{A}_q$ and $\mathcal{A}^{\varepsilon}_q$ (respectively). X_x stops at time $t_{\mathcal{A}_q}(x, q, \theta)$ while X^{ε}_x stops at time $t^{\varepsilon}_{\mathcal{A}^{\varepsilon}_q}(x, q, \theta^{\varepsilon})$.*

Moreover, we have

$$\left|J^\varepsilon(x,q;\theta^\varepsilon) - J(x,q;\theta)\right| < \varepsilon\left(K_\infty + \max\{K_\Phi, K_\Psi\}\right) \quad \forall \varepsilon \leq \bar{\varepsilon}$$

where

$$\begin{aligned} K_\infty &:= \frac{L_\ell}{\lambda - L_f} \\ K_\Phi &:= L_\Phi \frac{L_f}{\lambda} \\ K_\Psi &:= L_\Psi \left(\frac{L_f}{\lambda} + \frac{2}{\omega}(M_f + \lambda M_\ell + \lambda M_\Psi)\right) \end{aligned} \tag{A.3.2}$$

Proof. We will recover the estimate for the difference of the value functions w and w^ε by studying the difference of the corresponding cost functionals in the three cases. In order to simplify the notation, the dependency from the arguments t, x, q and u will be dropped when unnecessary.

1. From (A8), and since ℓ is Lipschitz continuous we have

$$\begin{aligned} &\left|J^\varepsilon(x,q;\theta^\varepsilon) - J(x,q;\theta)\right| \leq \\ &\leq \int_0^{+\infty} \left|\ell(X_x^\varepsilon + e, q, u) - \ell(X_x, q, u)\right| e^{-\lambda t} dt = \\ &= \int_0^{+\infty} \left|\ell(X_x^\varepsilon + e) - \ell(X_x^\varepsilon) + \ell(X_x^\varepsilon) - \ell(X_x)\right| e^{-\lambda t} dt \leq \\ &\leq \varepsilon L_\ell \int_0^{+\infty} e^{-\lambda t} dt + L_\ell \int_0^{+\infty} |X_x^\varepsilon - X_x| e^{-\lambda t} dt \leq \\ &\leq \varepsilon \frac{L_\ell}{\lambda} + \varepsilon L_\ell \int_0^{+\infty} e^{-(\lambda - L_f)t} dt - \varepsilon L_\ell \int_0^{+\infty} e^{-\lambda t} dt = \\ &= \varepsilon \frac{L_\ell}{\lambda - L_f} \end{aligned}$$

2. Since ℓ is bounded, for the previous result we have

$$\begin{aligned} &\left|J^\varepsilon(x,q;\theta^\varepsilon) - J(x,q;\theta)\right| \leq \\ &\leq \int_0^{\xi} \left|\ell(X_x^\varepsilon + e) - \ell(X_x)\right| e^{-\lambda t} dt + e^{-\lambda \xi} \left|\Phi(X_x^\varepsilon(\xi)) - \Phi(X_x(\xi))\right| \leq \\ &\leq \varepsilon \frac{L_\ell}{\lambda - L_f} + e^{-\lambda \xi} \left|\Phi(X_x^\varepsilon(\xi)) - \Phi(X_x(\xi))\right| \end{aligned}$$

   Using boundedness and Lipschitz continuity of Φ, the second term can be estimated

$$\begin{aligned} e^{-\lambda \xi} \left|\Phi(X_x^\varepsilon(\xi)) - \Phi(X_x(\xi))\right| &\leq L_\Phi e^{-\lambda \xi} |X_x^\varepsilon(\xi) - X_x(\xi)| \leq \\ &\leq \varepsilon L_\Phi \left(e^{-(\lambda - L_f)\xi} - e^{-\lambda \xi}\right) \leq \\ &\leq \varepsilon L_\Phi \left(1 - \frac{L_f}{\lambda}\right) \left(\frac{L_f}{\lambda - L_f}\right) \end{aligned}$$

Collecting the above results we finally obtain

$$\left|J^\varepsilon(x,q;\theta) - J(x,q;\theta)\right| < \varepsilon\left[\frac{L_\ell}{\lambda - L_f} + L_\Phi\left(1 - \frac{L_f}{\lambda}\right)\left(\frac{L_f}{\lambda - L_f}\right)\right]$$

3. By boundedness of ℓ, we have

$$\begin{aligned}
&\left|J^\varepsilon(x,q;\theta^\varepsilon) - J(x,q;\theta)\right| \leq \\
&\leq \int_0^{t^\varepsilon_{\mathcal{A}^\varepsilon_q}\wedge t_{\mathcal{A}_q}} \left|\ell(X^\varepsilon_x + e) - \ell(X_x)\right| e^{-\lambda t} dt + \left|\int_{t_{\mathcal{A}_q}}^{t^\varepsilon_{\mathcal{A}^\varepsilon_q}} \ell(X^\varepsilon_x + e) e^{-\lambda t} dt\right| + \\
&\quad + \left|e^{-\lambda t^\varepsilon_{\mathcal{A}^\varepsilon_q}} \Psi\big(X^\varepsilon_x(t^\varepsilon_{\mathcal{A}^\varepsilon_q})\big) - e^{-\lambda t_{\mathcal{A}_q}} \Psi\big(X_x(t_{\mathcal{A}_q})\big)\right| \leq \\
&\leq \varepsilon \frac{L_\ell}{\lambda - L_f} + \frac{M_\ell}{\lambda} e^{-\lambda(t^\varepsilon_{\mathcal{A}^\varepsilon_q}\wedge t_{\mathcal{A}_q})}\Big(1 - e^{-\lambda|t^\varepsilon_{\mathcal{A}^\varepsilon_q} - t_{\mathcal{A}_q}|}\Big) + \\
&\quad + \left|e^{-\lambda t^\varepsilon_{\mathcal{A}^\varepsilon_q}} \Psi\big(X^\varepsilon_x(t^\varepsilon_{\mathcal{A}^\varepsilon_q})\big) - e^{-\lambda t_{\mathcal{A}_q}} \Psi\big(X_x(t_{\mathcal{A}_q})\big)\right|
\end{aligned}$$

We will first obtain the estimate in terms of the quantities ε and $|t^\varepsilon_{\mathcal{A}^\varepsilon_q} - t_{\mathcal{A}_q}|$, then we will use Lemma A.3.3 to have a uniform bound depending only on ε.
The third term can be estimated using again the boundedness and Lipschitz continuity of Ψ:

$$\begin{aligned}
&\left|e^{-\lambda t^\varepsilon_{\mathcal{A}^\varepsilon_q}} \Psi\big(y^\varepsilon_x(t^\varepsilon_{\mathcal{A}^\varepsilon_q})\big) - e^{-\lambda t_{\mathcal{A}_q}} \Psi\big(X_x(t_{\mathcal{A}_q})\big)\right| \leq \\
&\quad \leq e^{-\lambda(t^\varepsilon_{\mathcal{A}^\varepsilon_q}\wedge t_{\mathcal{A}_q})}\left|\Psi\big(X^\varepsilon_x(t^\varepsilon_{\mathcal{A}^\varepsilon_q})\big) - \Psi\big(X_x(t_{\mathcal{A}_q})\big)\right| + \\
&\qquad + \Psi\big(X_x(t_{\mathcal{A}_q})\big) e^{-\lambda(t^\varepsilon_{\mathcal{A}^\varepsilon_q}\wedge t_{\mathcal{A}_q})}\left|e^{-\lambda t^\varepsilon_{\mathcal{A}^\varepsilon_q}} - e^{-\lambda t_{\mathcal{A}_q}}\right| \leq \\
&\quad \leq L_\Psi e^{-\lambda(t^\varepsilon_{\mathcal{A}^\varepsilon_q}\wedge t_{\mathcal{A}_q})}\left|X^\varepsilon_x(t^\varepsilon_{\mathcal{A}^\varepsilon_q}) - X_x(t_{\mathcal{A}_q})\right| + \\
&\qquad + M_\Psi e^{-\lambda(t^\varepsilon_{\mathcal{A}^\varepsilon_q}\wedge t_{\mathcal{A}_q})}\Big(1 - e^{-\lambda|t^\varepsilon_{\mathcal{A}^\varepsilon_q} - t_{\mathcal{A}_q}|}\Big)
\end{aligned}$$

where, if $t_{\mathcal{A}_q} \leq t^\varepsilon_{\mathcal{A}^\varepsilon_q}$ we have

$$\begin{aligned}
&e^{-\lambda(t^\varepsilon_{\mathcal{A}^\varepsilon_q}\wedge t_{\mathcal{A}_q})}\left|X^\varepsilon_x(t^\varepsilon_{\mathcal{A}^\varepsilon_q}) - X_x(t_{\mathcal{A}_q})\right| \leq \\
&\quad \leq e^{-\lambda t_{\mathcal{A}_q}}\Big(\left|X^\varepsilon_x(t^\varepsilon_{\mathcal{A}^\varepsilon_q}) - X^\varepsilon_x(t_{\mathcal{A}_q})\right| + \left|X^\varepsilon_x(t_{\mathcal{A}_q}) - X_x(t_{\mathcal{A}_q})\right|\Big) \leq \\
&\quad \leq e^{-\lambda t_{\mathcal{A}_q}}\left(M_f|t^\varepsilon_{\mathcal{A}^\varepsilon_q} - t_{\mathcal{A}_q}| + \varepsilon\Big(e^{-(\lambda - L_f)t_{\mathcal{A}_q}} - e^{-\lambda t_{\mathcal{A}_q}}\Big)\right) \leq \\
&\quad \leq e^{-\lambda t_{\mathcal{A}_q}} M_f|t^\varepsilon_{\mathcal{A}^\varepsilon_q} - t_{\mathcal{A}_q}| + \varepsilon\left(1 - \frac{L_f}{\lambda}\right)\left(\frac{L_f}{\lambda - L_f}\right)
\end{aligned}$$

Otherwise, if $t^{\varepsilon}_{\mathcal{A}^{\varepsilon}_q} < t_{\mathcal{A}_q}$

$$
\begin{aligned}
&e^{-\lambda(t^{\varepsilon}_{\mathcal{A}^{\varepsilon}_q}\wedge t_{\mathcal{A}_q})}\left|X^{\varepsilon}_x(t^{\varepsilon}_{\mathcal{A}^{\varepsilon}_q}) - X_x(t_{\mathcal{A}_q})\right| \leq \\
&\quad \leq e^{-\lambda t^{\varepsilon}_{\mathcal{A}^{\varepsilon}_q}}\left(\left|X_x(t^{\varepsilon}_{\mathcal{A}^{\varepsilon}_q}) - X_x(t_{\mathcal{A}_q})\right| + \left|X^{\varepsilon}_x(t^{\varepsilon}_{\mathcal{A}^{\varepsilon}_q}) - X_x(t^{\varepsilon}_{\mathcal{A}^{\varepsilon}_q})\right|\right) \leq \\
&\quad \leq e^{-\lambda t^{\varepsilon}_{\mathcal{A}^{\varepsilon}_q}} M_f |t^{\varepsilon}_{\mathcal{A}^{\varepsilon}_q} - t_{\mathcal{A}_q}| + \varepsilon\left(1 - \frac{L_f}{\lambda}\right)\left(\frac{L_f}{\lambda - L_f}\right)
\end{aligned}
$$

Hence, since $1 - e^{-x} \leq x$ we get

$$
\begin{aligned}
&\left|e^{-\lambda t^{\varepsilon}_{\mathcal{A}^{\varepsilon}_q}}\Psi\big(X^{\varepsilon}_x(t^{\varepsilon}_{\mathcal{A}^{\varepsilon}_q})\big) - e^{-\lambda t_{\mathcal{A}_q}}\Psi\big(X_x(t_{\mathcal{A}_q})\big)\right| \leq \\
&\quad \leq L_\Psi M_f e^{-\lambda(t^{\varepsilon}_{\mathcal{A}^{\varepsilon}_q}\wedge t_{\mathcal{A}_q})}|t^{\varepsilon}_{\mathcal{A}^{\varepsilon}_q} - t_{\mathcal{A}_q}| + \\
&\qquad + \varepsilon L_\Psi\left(1 - \frac{L_f}{\lambda}\right)\left(\frac{L_f}{\lambda - L_f}\right) + \\
&\qquad + \lambda L_\Psi M_\Psi e^{-\lambda(t^{\varepsilon}_{\mathcal{A}^{\varepsilon}_q}\wedge t_{\mathcal{A}_q})}|t^{\varepsilon}_{\mathcal{A}^{\varepsilon}_q} - t_{\mathcal{A}_q}|
\end{aligned}
$$

By Lemma A.3.3, we can apply the following inequality

$$
\begin{aligned}
e^{-\lambda(t^{\varepsilon}_{\mathcal{A}^{\varepsilon}_q}\wedge t_{\mathcal{A}_q})}|t^{\varepsilon}_{\mathcal{A}^{\varepsilon}_q} - t_{\mathcal{A}_q}| &< \varepsilon\frac{2}{\omega}e^{-\lambda(t^{\varepsilon}_{\mathcal{A}^{\varepsilon}_q}\wedge t_{\mathcal{A}_q})}e^{L_f(t^{\varepsilon}_{\mathcal{A}^{\varepsilon}_q}\wedge t_{\mathcal{A}_q})} \leq \\
&\leq e^{-(\lambda - L_f)(t^{\varepsilon}_{\mathcal{A}^{\varepsilon}_q}\wedge t_{\mathcal{A}_q})} \leq \varepsilon\frac{2}{\omega}
\end{aligned}
$$

and conclude

$$
\begin{aligned}
&\left|e^{-\lambda t^{\varepsilon}_{\mathcal{A}^{\varepsilon}_q}}\Psi\big(X^{\varepsilon}_x(t^{\varepsilon}_{\mathcal{A}^{\varepsilon}_q})\big) - e^{-\lambda t_{\mathcal{A}_q}}\Psi\big(X_x(t_{\mathcal{A}_q})\big)\right| < \\
&\quad < \varepsilon L_\Psi\left(1 - \frac{L_f}{\lambda}\right)\left(\frac{L_f}{\lambda - L_f}\right) + \varepsilon L_\Psi M_f\frac{2}{\omega} + \varepsilon\lambda L_\Psi M_\Psi\frac{2}{\omega}
\end{aligned}
$$

Collecting the results above, we finally obtain

$$
\begin{aligned}
|J^{\varepsilon}(x,q;\theta^{\varepsilon}) - J(x,q;\theta)| < \varepsilon\Bigg[&\frac{L_\ell}{\lambda - L_f} + L_\Psi\left(1 - \frac{L_f}{\lambda}\right)\left(\frac{L_f}{\lambda - L_f}\right) + \\
&+ \frac{2}{\omega}L_\Psi\left(M_f + \lambda M_\ell + \lambda M_\Psi\right)\Bigg]
\end{aligned}
$$

□

A.3.3 Estimate for the value functions

We are finally able to estimate the error between the solutions of problems (4.3.5) and (4.2.1).

Theorem A.3.1. *If (A1)-(A7) hold, then there exists $\bar{\varepsilon} > 0$ such that, for every $x \in \mathbb{R}^d \setminus (\mathcal{A}_q \cup \mathcal{C}_q)$, $q \in \mathbb{I}$, $\delta > 0$ and $\varepsilon \leq \bar{\varepsilon}$ we have*

$$\left|w(x,q) - w^\varepsilon(x,q)\right| < \varepsilon K_{\Phi,\Psi} + \delta$$

where $K_{\Phi,\Psi} := K_\infty + \max\{K_\Phi, K_\Psi\}$, with K_∞, K_Φ and K_Φ as defined in (A.3.2)

Proof. From the definition of the value functions w and w^ε we know that for each $\delta > 0$ there exist θ_δ and $\theta^\varepsilon_\delta$ respectively in Θ and Θ^ε such that

$$J(x,q;\theta_\delta) \leq w(x,q) + \delta$$
$$J^\varepsilon(x,q;\theta^\varepsilon_\delta) \leq w^\varepsilon(x,q) + \delta$$

then, by Proposition A.3.1, it follows

$$w(x,q) - w^\varepsilon(x,q) \leq J(x,q;\theta_\delta) - J^\varepsilon(x,q;\theta^\varepsilon_\delta) + \delta < \varepsilon K_{\Phi,\Psi} + \delta.$$

On the other hand, we also have $0 \leq w(x,q) - w^\varepsilon(x,q)$ because $\Theta \subset \Theta^\varepsilon$, and by combining the two inequalities we obtain our result. □

A.3.4 Estimate on the perturbed numerical approximation

As we did in the previous section, we will drop the assumptions $\mathcal{C} = \mathbb{R}^d$ and $\mathcal{A} = \emptyset$ made in 4.0.1 and consider the more general case in which the two sets just non-empty.

We want to examine here the difference between the numerical approximations of respectively the QVI with a constant obstacles and its perturbed version in the case of a Semi-Lagrangian scheme.

We recall that the non-perturbed system is

$$\begin{cases} \lambda w(x,q) + H\big(x,q,D_x w(x,q)\big) \leq 0 & \text{on } \mathbb{R}^d \setminus (\mathcal{A} \cup \mathcal{C}) \quad \text{(A.3.3a)} \\ \max\Big\{\lambda w(x,q) + H\big(x,q,D_x w(x,q)\big), \\ \qquad w(x,q) - \Phi(x,q)\Big\} \leq 0 & \text{on } \mathcal{C} \quad \text{(A.3.3b)} \\ w(x,q) - \Psi(x,q) = 0 & \text{on } \mathcal{A}. \quad \text{(A.3.3c)} \end{cases}$$

It can be approximated with the scheme

$$W_h(x,q) = \Theta^h(x,q,W_h) := \begin{cases} \Sigma^h(x,q,W_h) & x \in \mathbb{R}^d \setminus (\mathcal{A} \cup \mathcal{C}) \\ \min\big\{\Sigma^h(x,q,W_h), \\ \qquad \Phi(x,q)\big\} & x \in \mathcal{C} \\ \Psi(x,q) & x \in \mathcal{A}. \end{cases} \quad \text{(A.3.4)}$$

The perturbed SL scheme is obtained by replacing Σ^h in (A.3.4) with the mapping

$$\Sigma^{\varepsilon,h}(x,q,W_h^\varepsilon) = \\ = \min_{u\in U,|e|\leq\varepsilon}\left\{h\ell(x+e,q,u)+(1-\lambda h)\mathcal{I}[W_h^\varepsilon](x+hf(x+e,q,u),q)\right\}. \tag{A.3.5}$$

We start by giving the following general result:

Theorem A.3.2. *Let (A1)-(A7) and (S1)-(S4) hold, and let W_h and W_h^ε be respectively solution of* (A.3.4) *and its perturbed version* (A.3.5) *with Φ finite or infinite. Then, the perturbed SL scheme has a unique bounded and uniformly Lipschitz continuous solution W_h^ε.*

Proof. It suffices to note that, with the addition of the term e, the problem still satisfies the basic assumptions, and all the relevant constants of the problem remain unchanged. Then, the result follows from Theorem A.2.1, implying

$$|W_h^\varepsilon|_1 \leq (1+(\lambda-L_f)h)\max\left\{L_\Phi, L_\Psi, \frac{L_\ell}{\lambda-L_f}\right\}.$$

□

Let now W_h^ε denote the numerical solution for the perturbed SL scheme. We prove the following.

Theorem A.3.3. *Let (A1)-(A7) and (S1)-(S4) hold, and let W_h and W_h^ε be respectively solution of* (A.3.4) *and its perturbed version* (A.3.5) *with Φ finite or infinite. Then, for ε and h small enough, we have*

$$|W_h - W_h^\varepsilon|_0 \leq \varepsilon K_{W_h,h} \tag{A.3.6}$$

with

$$K_{W_h,h} := \max\left\{(L_\ell + L_{W_h}L_f)h, \frac{L_\ell + L_{W_h}L_f}{\lambda}\right\}.$$

Proof. We recall that both the exact and the approximate solutions for either the original or the perturbed problem are Lipschitz continuous.

Using a scheme in fixed point SL form, the unperturbed QVI is approximated by (A.3.4), whereas its perturbed version is given by

$$W_h^\varepsilon(x,q) = \Theta^{\varepsilon,h}(x,q,W_h^\varepsilon) := \begin{cases} \min\left\{\Sigma^{\varepsilon,h}(x,q,W_h^\varepsilon), \Phi(x,q)\right\} & \text{on } \mathcal{C}^\varepsilon \\ \Psi(x,q) & \text{on } \mathcal{A}^\varepsilon \\ \Sigma^{\varepsilon,h}(x,q,W_h^\varepsilon) & \text{else} \end{cases} \tag{A.3.7}$$

The idea is to apply the two schemes to Lipschitz continuous numerical solutions W_h and W_h^ε and estimate the difference

$$\begin{aligned}|T^h(\,\cdot\,,\cdot\,,W_h) - T^{\varepsilon,h}(\,\cdot\,,\cdot\,,W_h^\varepsilon)|_0 \le |T^h(\,\cdot\,,\cdot\,,W_h) - T^h(\,\cdot\,,\cdot\,,W_h^\varepsilon)|_0 + \\ + |T^h(\,\cdot\,,\cdot\,,W_h^\varepsilon) - T^{\varepsilon,h}(\,\cdot\,,\cdot\,,W_h^\varepsilon)|_0\end{aligned} \tag{A.3.8}$$

Using now, for $T = \Theta^h, \Theta^{\varepsilon,h}, \Sigma^h, \Sigma^{\varepsilon,h}$ and $U = W_h, W_h^\varepsilon$, the shorthand notation

$$T(U) := T(\,\cdot\,,\cdot\,,U),$$

we can single out four cases:

a) $(x,q) \in (\mathbb{I}\times\mathbb{R}^d)\setminus(\mathcal{A}\cup\mathcal{C})$.

In this case, we can obtain for the first term in (A.3.8)

$$|\Theta^h(W_h) - \Theta^h(W_h^\varepsilon)|_0 = |\Sigma^h(W_h) - \Sigma^h(W_h^\varepsilon)|_0 \le (1-\lambda h)|W_h - W_h^\varepsilon|_0 \tag{A.3.9}$$

whereas for the second, there exists $C>0$ such that:

$$|T^h(W_h^\varepsilon) - T^{\varepsilon,h}(W_h^\varepsilon)|_0 \le Ch\varepsilon. \tag{A.3.10}$$

The former inequality is a known property of the SL scheme, while the latter is easily obtained by considering that the Lipschitz continuity of ℓ and f, along with the bound on $|e|$, imply that

$$\begin{aligned}|\ell(x,q,u) - \ell(x+e,q,u)| \le L_\ell\varepsilon,\\ |f(x,q,u) - f(x+e,q,u)| \le L_f\varepsilon\end{aligned}$$

so that, taking into account the Lipschitz continuity of W_h, by a standard argument we obtain (A.3.10) as

$$|\Sigma^h(W_h) - \Sigma^{\varepsilon,h}(W_h)|_0 \le (L_\ell + L_{W_h}L_f)h\varepsilon$$

b) $(x,q)\in\mathcal{A}^\varepsilon$.

In this case, we recall that $\mathcal{A}^\varepsilon\subset\mathcal{A}$, which clearly implies that

$$\Theta^h(x,q,W_h) = \Theta^{\varepsilon,h}(x,q,W_h^\varepsilon) = \Psi(x,q)$$

and the left-hand side of (A.3.8) vanishes.

c) $(x,q)\in\mathcal{C}^\varepsilon$.

First, note that in case the minimum in both $\Theta^h(W_h)$ and $\Theta^h(W_h^\varepsilon)$ is attained by the same operator, then there is nothing else to prove. In fact, in this case the estimates (A.3.9)-(A.3.10) are recovered by mixing the arguments of the previous cases a) and b). Let therefore the

min be achieved by different operators, e.g., let $\Theta^h(W_h) = \Sigma^h(W_h)$ and $\Theta^h(W_h^\varepsilon) = \Phi$. Working in terms of unilateral estimates, we have

$$\begin{aligned}\Theta^h(x,q,W_h) - \Theta^h(x,q,W_h^\varepsilon) &= \Sigma^h(x,q,W_h) - \Phi(x,q) \leq \\ &\leq \Phi(x,q) - \Phi(x,q) = 0\end{aligned}$$

in which we get the inequality by replacing the argmin in $\Theta^h(W_h)$ with the other choice. In a parallel form, we obtain the reverse inequality, as

$$\begin{aligned}\Theta^h(x,q,W_h^\varepsilon) - \Theta^h(x,q,W_h) &= \Phi(x,q) - \Sigma^h(x,q,W_h) \\ &\leq \Sigma^h(x,q,W_h^\varepsilon) - \Sigma^h(x,q,W_h) \\ &\leq (1-\lambda h)|W_h - W_h^\varepsilon|_0\end{aligned}$$

The same arguments can then be applied to the case in which the choice of the operators is reversed, so that we finally obtain (A.3.9). Note that by a further straightforward adaptation of this technique it is also possible to obtain (A.3.10). We leave this part of the proof to the reader.

Moreover, we can observe that, due to the transversality condition (A5), both $\mathcal{A}$ and $\mathcal{C}$ (and the perturbed sets if ε is small enough) are invariant for the dynamics and (if h is also small enough) for the scheme. In particular, for $(x,q) \in \mathcal{C}^\varepsilon$, we can understand the ∞-norm in (A.3.9)–(A.3.10) as the norm on the set $\mathcal{C}^\varepsilon$ itself. We obtain therefore, by iterating the estimate (A.3.8) in (A.3.4) and (A.3.7) from the same initial guess $W_h^{(0)} = W_h^{\varepsilon(0)}$:

$$|W_h - W_h^\varepsilon|_0 \leq (L_\ell + L_{W_h} L_f)\varepsilon h \sum_{k\geq 0}(1-\lambda h)^k = \frac{L_\ell + L_{W_h} L_f}{\lambda}\varepsilon$$

d) $(x,q) \in \mathcal{C} \setminus \mathcal{C}^\varepsilon$ or $(x,q) \in \mathcal{A} \setminus \mathcal{A}^\varepsilon$

In this last case, due to the transversality condition (A5), we can apply the same arguments of the previous case to extend the estimate obtained on $\mathcal{A}^\varepsilon$ and $\mathcal{C}^\varepsilon$ to the whole of respectively $\mathcal{A}$ and $\mathcal{C}$ by adding a term $O(\varepsilon)$.

We can therefore conclude by collecting all the cases above in the bound

$$|W_h - W_h^\varepsilon|_0 \leq \max\left\{(L_\ell + L_{W_h} L_f)h, \frac{L_\ell + L_{W_h} L_f}{\lambda}\right\}\varepsilon.$$

□

Chapter 5

Conclusions

In this first part of the thesis we showed how the numerical solution of hybrid optimal control problems can be analyzed from an abstract and more general point of view, leading to the main result on the error estimate between the value function and its approximation, under suitable assumptions. We also have constructed and validated a Semi-Lagrangian scheme for hybrid Dynamic Programming problems in infinite horizon form, improving its efficiency with the application of a Policy Iteration type algorithm.

We started by describing the general framework of this kind of problems, listing and detailing all the necessary hypotheses to be used later. The Dynamic Programming Principle allowed us to characterize the value function of the hybrid optimal control problem as the solution of a particular Quasi-Variational Inequality.

Then we showed how to construct and solve a Semi-Lagrangian type scheme for the approximation of the value function and the corresponding optimal control strategy. The scheme presented not only satisfies all the convergence hypotheses, but can also be made more efficient by applying the Policy Iteration algorithm, as opposed to the Value Iteration, a more traditional solution method. Numerical tests performed on examples of varying complexity show that the scheme is robust and that the approximate optimal control policy obtained is stable and accurate, although the complexity remains critical with respect to the dimension of the state space. This validation suggests that this could be a feasible method to design optimization-based static controllers in low dimension.

On the theoretical point of view, the QVI characterizing the value function also allows us to define the starting point of the method used to obtain some the error estimates: the cascade. This technique allowed us to approach the original hybrid control problem by studying a sequence of less complex obstacle problems, providing us with a chain of error estimates which, combined, give us the main result. It's important to remark that we use used the cascade as a tool for obtaining the theoretical result, and not

for actually solving the hybrid control problem numerically. We also point out that some of the assumptions made throughout this part are not true a priori for every numerical scheme, such as the existence, the uniqueness or the Lipschitz continuity of its solution. Nonetheless, we make sure that they hold at least for SL schemes.

Our results have been obtained under standard hypotheses that in some more realistic scenarios might not hold. For this reason, further development of this theory could involve the extension of the results presented to a more general framework for Hybrid Systems. Moreover, it could be interesting to study the numerical approximation of such systems with numerical methods which don't satisfy all the properties we required, such as the non-monotone Essentially Non-Oscillatory (ENO) scheme, in order to obtain the same error estimates for a wider variety of schemes.

Part II

Chance-constrained optimization in aerospace

Chapter 6

The chance-constrained optimization problem

6.1 Introduction

One of the earliest and most famous examples of optimization problems in aerospace dates back to the beginning of the twentieth century. In 1921, American physicist Robert Hutchings Goddard published a paper titled "A Method of Reaching Extreme Altitudes" in which he studied the problem of minimizing the fuel consumption of a rocket ascending vertically from Earth's surface, taking into account both atmospheric drag and gravitational field.

In order to better explain the nature of this kind of problems, we give a simplified formulation of the one studied by Goddard, but first we need to formally define what an optimal control problem is. We start by recalling the definition of a *controlled system*. A controlled system usually consists in a set of ordinary differential equations parameterized by a function called *control*:

$$\begin{cases} \dot{y}(t) = f\big(t, y(t), u(t)\big) & \forall t \in (0, t_{\mathrm{f}}] \\ y(0) = y_0 \end{cases} \tag{6.1.1}$$

where $f : \mathbb{R}_+ \times \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^n$ is the *state* function, $u : \mathbb{R}_+ \to U \subset \mathbb{R}^m$ is the *control function* and the final time t_{f} can be either finite or infinite. A control u is said to be *admissible* if it belongs to a given set $\mathcal{U}$, and each u in the admissible set selects a *trajectory* $y_u : \mathbb{R}_+ \to \mathbb{R}^n$.

What characterizes an optimal control problem is the presence of a *cost functional*, whose purpose is to measure the quality of a control strategy with respect to a chosen criterion. For a controlled system in the form of (6.1.1), the cost functional $J : \mathcal{U} \to \mathbb{R}_+$ is defined as

$$J(u) := \phi\big(t_{\mathrm{f}}, y(t_{\mathrm{f}})\big) + \int_0^{t_{\mathrm{f}}} \ell\big(t, y(t), u(t)\big) dt \tag{6.1.2}$$

where the functionals $\phi : \mathbb{R}_+ \times \mathbb{R}^n \to \mathbb{R}_+$ and $\ell : \mathbb{R}_+ \times \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}_+$ are called *final cost* and *running cost* respectively. Since we are interested in finding an *optimal control strategy* u^* that minimizes the cost (6.1.2), we define our *optimal control problem* as

$$\min_{u \in \mathcal{U}} J(u).$$

Sometimes we might require the system trajectory y to satisfy some other constraints in addition to the ODE system (6.1.1). In this case we can generalize the previous formulation by adding *constraint functions*:

$$\begin{cases} \min\limits_{u \in \mathcal{U}} J(u) \\ G(t, y(t), u(t)) \geq 0 \quad \forall t \in (0, t_\mathrm{f}] \\ H(t, y(t), u(t)) = 0 \quad \forall t \in (0, t_\mathrm{f}] \end{cases} \tag{6.1.3}$$

where $G : \mathbb{R}_+ \times \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^l$ represents the set of *inequality constraints* and $H : \mathbb{R}_+ \times \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^p$ the set of *equality constraints*.

Going back to our example, consider the vertical ascent of a rocket in one dimension. At a given time t, we measure the position of the launcher by means of its altitude $r(t)$ and its speed with the scalar value $v(t)$. We introduce the variable $u(t) \in [0, 1]$ which represents the percentage of maximum thrust applied at a given time. The vehicle starts from a still position at ground level. At time $t = 0$ the thrust force $Tu(t)$ of the engine pushes the launcher upwards, against the force of gravity $m(t)g$. $m(t)$ denotes the mass of the launcher at time t, and it is consumed at the rate $\frac{T}{v_\mathrm{e}} u(t)$. We consider the maximum thrust T, the fuel speed v_e, the initial mass m_0 and the final time t_f to be fixed. The controlled system describing the launcher's dynamics is

$$\begin{cases} \dot{r}(t) = v(t) & t \in (0, t_\mathrm{f}] \\ \dot{v}(t) = \frac{T}{m(t)} u(t) - g & t \in (0, t_\mathrm{f}] \\ \dot{m}(t) = -\frac{T}{v_\mathrm{e}} u(t) & t \in (0, t_\mathrm{f}] \\ r(0) = 0 \\ v(0) = 0 \\ m(0) = m_0 \end{cases} \tag{6.1.4}$$

where the admissible control set is

$$\mathcal{U} := \{u : \mathbb{R}_+ \to [0, 1] \subset \mathbb{R} \quad | \quad u \text{ is measurable}\}.$$

We want to solve the optimal control problem of finding a particular $u^* \in \mathcal{U}$ that maximizes the final mass of the launcher while making sure it reaches at least a given altitude $\bar{r}_\mathrm{f}$ at time t_f. Formally, this translates to solving

$$\begin{cases} \max\limits_{u \in \mathcal{U}} m_\mathrm{f}(u) \\ r_\mathrm{f}(u) \geq \bar{r}_\mathrm{f} \end{cases} \tag{6.1.5}$$

where cost $m_{\mathrm{f}}(u)$ and constraint $r_{\mathrm{f}}(u)$ functions are represented by respectively final mass $m(t_{\mathrm{f}})$ and altitude $r(t_{\mathrm{f}})$ corresponding to the control u.

In the particular case of this thesis, we are interested in the specific field of parametric optimization, that is, a class of optimization problems characterized by the fact that the optimal value of the cost function depends on one or more parameters defining the mathematical model. Drawing a parallel with the previous example, let us suppose that we want to study how the optimal final mass of the launcher changes with respect to the maximum thrust T. We explicit the dependency of the cost and constraint functions on the parameter and denote the optimal cost for a given T with $m^*(T)$:

$$\begin{cases} m^*(T) := \max\limits_{u \in \mathcal{U}} m_{\mathrm{f}}(u, T) \\ r_{\mathrm{f}}(u, T) \geq \bar{r}_{\mathrm{f}} \end{cases}$$

This formulation has many applications, for example, it can be used for studying the sensitivity of $m^*(T)$ with respect to the parameter T, allowing us to predict how the final mass of the launcher would respond to changes in the engine's thrust.

Since we are interested in the case where T is subject to unpredictable variations, we represent this behavior by redefining T as a random variable taking values inside an interval $[T_-, T_+]$, according to a given probability distribution. As a consequence of this definition, the two functions $m_{\mathrm{f}}(u, T)$ and $r_{\mathrm{f}}(u, T)$ also become random variables, thus forcing us to reformulate the problem accordingly. For this purpose we introduce the parameter $p \in [0, 1]$ and consider the following problem

$$\begin{cases} \max\limits_{u \in \mathcal{U}} \mathbb{E}\big[m_{\mathrm{f}}(u, T)\big] \\ \mathbb{P}\big[r_{\mathrm{f}}(u, T) \geq \bar{r}_{\mathrm{f}}\big] \geq p \end{cases}$$

where $\mathbb{E}$ denotes the expectation and $\mathbb{P}$ the probability. Here p acts as a probability threshold for the realization of the event $r_{\mathrm{f}}(u, T) \geq r_{\mathrm{f}}$, and the inequality $\mathbb{P}\big[r_{\mathrm{f}}(u, T) \geq \bar{r}_{\mathrm{f}}\big] \geq p$ is called *chance* or *probability* constraint.

The resulting problem belongs to the subclass of stochastic optimization problems known as chance-constrained optimization problems. The goal of this chapter is to show how Lagrangian methods and non-parametric statistics can be used to solve this kind of problems, focusing on the details of the algorithmic approach. We analyze two distinct methods: the Stochastic Arrow-Hurwicz Algorithm (SAHA) and the Kernel Density Estimation (KDE). The former is a combination of the Monte Carlo method and the iterative gradient method, while the latter is a technique used to approximate the probability density function of a random variable with unknown distribution, from a relatively small sample. We explain how these methods can be implemented numerically and applied to several examples of chance-constrained optimization in aerospace.

The rest of this introduction will focus on the definition of the type of chance-constrained optimization problems we will solve, presenting the techniques that will be adopted later. In Section 6.2 we mention some existing results on the subject of chance-constrained optimization, while Section 7.1.2 gives an overview of the SAHA and KDE techniques. Section 7.2 consists in a collection six numerical examples involving the application of KDE to chance-constrained optimization and optimal control problems.

6.1.1 The chance-constrained optimization problem

Before describing the type of chance-constrained optimization problems we are interested in, we will give a brief overview of the history and the results of parametric optimization.

Using example (6.1.5) as a reference, we start by considering the formal definition of a parametric optimization problem given in [14]:

$$\begin{cases} \min\limits_{x \in \mathcal{X}} J(x, \xi) \\ G(x, \xi) \in \mathcal{G}. \end{cases} \tag{6.1.6}$$

In the general case, $\mathcal{X}$ is a Banach space and the parameter ξ can be a scalar, a vector or an element of an appropriate normed or metric space Ω. The set $\mathcal{G}$ is a convex subset of a Banach space $\mathcal{Y}$ and $G : \mathcal{X} \times \Omega \to \mathcal{Y}$.

A crucial aspect of parametric optimization is the notion of stability. It can be used to retrieve many information on the regularity of the value function with respect to ξ, and obtain error estimates. For this reason, we report one of the most important definitions of stability. If there exists $(x^*, \xi^*) \in \mathcal{X} \times \Omega$ such that the following regularity condition is satisfied

$$0 \in \operatorname{int}\{G(x, \xi) + D_x G(x, \xi)\mathcal{X} - \mathcal{G}\}$$

then, for all (x, ξ) in a neighborhood of (x^*, ξ^*), from [14, Proposition 3.3] we have

$$d\Big(x, \{x \in \mathcal{X} \quad | \quad G(x, \xi) \in \mathcal{G}\}\Big) = O\Big(d(G(x, \xi), \mathcal{G})\Big) \tag{6.1.7}$$

where $d(x, \mathcal{Y}) := \inf_{y \in \mathcal{Y}} \|x - y\|$ and D_x denotes the partial derivative with respect to x. Property (6.1.7) is called *metric regularity*, and it can be used for obtaining upper estimates for the optimal value function.

The study of this category of problems can be traced back to the works of Chebyschev on uniform approximations by algebraic polynomials, but for an important development of the subject we have to wait until the '60s and '70s, thanks to the mathematicians who worked in parallel between the former Soviet Union (e.g. Levitin [43, 44]) and the Western school (e.g. Danskin, Dem'yanov and Malozemov [23, 24]).

Robust optimization

An important subject in parametric optimization is robust optimization. This approach makes use of the worst-case analysis to treat uncertainties in order to obtain what is called a "robust" solution, i.e. a solution whose worst outcome is equal or better than the ones of all the other feasible solutions. Using a variation of Wald's maximin model (see [70]), we consider the following mathematical formulations of a robust optimization problem.

$$\begin{cases} \min\limits_{x\in\mathcal{X}} \max\limits_{\xi\in\Omega} J(x,\xi) \\ G(x,\xi) \in \mathcal{G} \qquad\qquad \forall \xi \in \Omega. \end{cases} \tag{6.1.8}$$

This model is named after Abraham Wald, who developed it in the 1940s, taking inspiration from similar models used in game theory. He studied this approach while attempting to solve problems involving one agent playing, figuratively, with a pessimistic attitude towards all possible outcomes. Despite the presence of uncertain parameters, robust optimization is a deterministic method and it doesn't involve probability or random variables. That is a feature of what is called stochastic optimization, and it will be discussed later.

In parallel with the example, we can apply the robust optimization approach to problem (6.1.5). By defining $x := u$, $\mathcal{X} := \mathcal{U}$, $J(x,\xi) := -m_{\mathrm{f}}(u,T)$, $G(x,\xi) := r_{\mathrm{f}}(u,T) - \bar{r}_{\mathrm{f}}$ and $\mathcal{G} := \mathbb{R}_+$, we obtain what is called a *robust optimal control problem*:

$$\begin{cases} \max\limits_{u\in\mathcal{U}} \min\limits_{T\in[T_-,T_+]} m_{\mathrm{f}}(u,T) \\ r_{\mathrm{f}}(u,T) \geq \bar{r}_{\mathrm{f}} \qquad\qquad \forall T \in [T_-,T_+]. \end{cases} \tag{6.1.9}$$

A solution to this system is a control strategy $u^* \in \mathcal{U}$ that maximizes the final mass of the launcher even for the worst realization of the parameter T, while satisfying the constraint on the final altitude for any $T \in [T_-,T_+]$.

As pointed out in [10], robust optimization requires a trade-off: the price for obtaining a solution that is feasible in every scenario often results in the suboptimality of the value function. Moreover, there might exist problems in which the constraint function cannot be satisfied for every realization of the model's parameters. In the case of our example (6.1.9), let us look at the controlled system (6.1.4) describing the dynamics. In order for the launcher to take off, its thrust T and initial mass m_0 must satisfy the relation $m_0 \leq \frac{T}{g}$, otherwise the engine wouldn't be powerful enough to lift the weight of the rocket. If, for example, we assign to the initial mass the value 8.75 and choose a gravitational acceleration g equal to 9.8, we have

$$\frac{T}{g} \geq m_0 \iff T \geq m_0 g = 8.75 \cdot 9.8 = 85.75.$$

This means that if T varies in the interval $[T_-, T_+] = [80, 90]$, we can't apply the robust approach defined in (6.1.9), since there exist values of T inside $[T_-, T_+]$ for which the launcher can't take off and thus it can't possibly reach the required final altitude $\bar{r}_\mathrm{f}$.

One way for addressing this kind of issues is constraint relaxation: in the general formulation (6.1.8), instead of asking the constraint $G(x, \xi) \in \mathcal{G}$ to be satisfied for every $\xi \in \Omega$, we can substitute this requirement with less strict ones. There are many ways to relax this kind of constraints, and in the case of optimal control problems we report the controllability approach illustrated in [1]. Consider the parameterized control system

$$\begin{cases} \dot{y}_\xi = f_\xi(y_\xi(t), u(t)) & t \in (0, t_\mathrm{f}] \\ y_\xi(0) = y_0(\xi) \end{cases}$$

where the state y_ξ of the system belongs to a finite-dimensional manifold $\mathcal{M}$. The set of admissible controls is

$$\mathcal{U} := \{u : \mathbb{R}_+ \to U \subset \mathbb{R}^q \quad | \quad u \text{ is measurable}\}$$

and $\xi \mapsto f_\xi$ is a family of vector fields on $\mathcal{M}$ parameterized by $\xi \in \Omega \subseteq \mathbb{R}^m$. The system is said to be *L_q-approximately controllable* in time t_f from $y_0(\xi)$ to $y_\mathrm{f}(\xi)$ if for every $\delta > 0$ there exists a ξ-independent control u such that

$$\left|y_\xi(t_\mathrm{f}) - y_\mathrm{f}(\xi)\right|_{L_q} < \delta$$

where the function $y_\mathrm{f}(\xi)$ is called target and $|\cdot|_{L_q}$ denotes the usual L_q norm. We refer to [1] for the requirements that guarantee this property. We also point out that, although the authors only use the canonical Lebesgue measure, it could be worth exploring the introduction of a probability measure on the set of parameters.

We can use problem (6.1.5) to show an application of L_q-approximate controllability. In our case the role of the parameter ξ is covered by T and the system's state and initial conditions are denoted by

$$\begin{aligned} y_T &:= (r(t,u), v(t,u), m(t,u)) \\ f_T(y_T(t), u(t)) &:= \left(v(t,u), \frac{T}{m(t,u)}u(t) - g, -\frac{T}{v_\mathrm{e}}u(t)\right) \\ y_0 &:= (0, 0, m_0). \end{aligned}$$

If our system is L_q-approximately controllable, we can fix $\delta > 0$ and $q > 0$, and replace the constraint in (6.1.9) with the controllability condition to obtain

$$\begin{cases} \max\limits_{u \in \mathcal{U}} \min\limits_{T \in [T_-, T_+]} m_\mathrm{f}(u, T) \\ \left|\min\left\{r_\mathrm{f}(u, T) - \bar{r}_\mathrm{f}, 0\right\}\right|_{L_q} < \delta \quad \forall T \in [T_-, T_+]. \end{cases}$$

This means that we are willing to tolerate an error δ for the violation of the constraint $r_{\mathrm{f}}(u,T) - \bar{r}_{\mathrm{f}}$ with respect to the L_q norm. The relaxation of the constraint allows for more control strategies u to be considered, implying that the corresponding optimal value for the final mass is bigger than the one we would obtain with the more strict formulation stated in (6.1.9). We refer to [45, 62] as well as to two other Ph.D. theses [7, 61] as sources for other approaches in the field of robust optimization in aerospace.

Chance-constrained optimization

We already mentioned that the robust approach to parametric optimization comes with some disadvantages: it might be difficult to guarantee the existence of a solution due to the strictness of the constraint in (6.1.8), and even in the case of a relaxation approach, it's hard to make sure that the problem satisfies all the required controllability hypotheses.

Chance constraints can be seen as another method for relaxing the constraint in (6.1.8). The basic idea is to choose a probability distribution for the parameter ξ and treat it as a random variable. Consider the following formulation for a chance-constrained optimization problem:

$$\begin{cases} \min\limits_{x\in\mathcal{X}} \mathbb{E}\big[J(x,\xi)\big] \\ \mathbb{P}\big[G(x,\xi)\geq 0\big] \geq p \end{cases} \tag{6.1.10}$$

where $\mathcal{X}\subseteq\mathbb{R}^n$ is the admissible set for the decision variables in x, $J:\mathbb{R}^n\to\mathbb{R}$ is an objective, $G:\mathbb{R}^n\times\mathbb{R}^m\to\mathbb{R}$ defines an inequality, $p\in(0,1)$ is a probability threshold called *confidence level* and ξ is an m-dimensional random vector defined on some probability space $(\Omega,\mathcal{A},\mathbb{P})$. The probability distribution of ξ will be denoted by $\mu := \mathbb{P}\circ\xi^{-1}\in\mathcal{P}(\mathbb{R}^m)$, where $\mathcal{P}(\mathbb{R}^m)$ is the space of Borel probability measures on $\mathbb{R}^m$.

This kind of problem has been treated at least since the fifties with the work of Charnes, Cooper and Symonds (see [21]). A general theory, however, is due to Prékopa in [51, 52], who also introduced the convexity theory based on logconcavity. Other contributions on logconcavity theory in stochastic programming can be found in [53, 54, 25].

There are many variations to the chance-constrained optimization problem. For example, when considering multiple constraints, they might be either *joint* or *disjoint*. Let $l>1$ be the number of constraints, with $G_i:\mathbb{R}^n\times\mathbb{R}^m\to\mathbb{R}$ denoting the function defining the i-th constraint. In the case of joint constraints, the problem becomes

$$\begin{cases} \min\limits_{x\in\mathcal{X}} \mathbb{E}\big[J(x,\xi)\big] \\ \mathbb{P}\Big[G_i(x,\xi)\geq 0 \quad \forall i\in\{1,2,\dots,l\}\Big] \geq p \end{cases} \tag{6.1.11}$$

implying that all the inequalities $\mathbb{P}\big[G_i(x,\xi)\geq 0\big]\geq p$ must hold with the same confidence level p. On the other hand, disjoint chance constraints

require a different confidence level $p_i \in (0,1)$ for each inequality:

$$\begin{cases} \min\limits_{x\in\mathcal{X}} \mathbb{E}\big[J(x,\xi)\big] \\ \mathbb{P}\big[G_i(x,\xi) \geq 0\big] \geq p_i \quad \forall i \in \{1,2,\dots,l\}. \end{cases}$$

Although there are many other ways of formulating a chance-constrained optimization problem, for the purpose of our tests we are mainly interested in two of them. The first one is a slightly simpler version of (6.1.10), in which the cost function does not depend on the random array ξ:

$$\begin{cases} \min\limits_{x\in\mathcal{X}} J(x) \\ \mathbb{P}\big[G(x,\xi) \geq 0\big] \geq p. \end{cases}$$

The second formulation is the one used in the particular case of *percentile optimization*, where there is only one decision variable and one constraint, and the cost function is the decision variable itself:

$$\begin{cases} \min\limits_{\mu\in\mathbb{R}} \mu \\ \mathbb{P}\big[G(\xi) \leq \mu\big] \geq p \end{cases} \tag{6.1.12}$$

The name comes from the fact that this problem is aimed at finding μ such that μ is the the p-percentile of the distribution of $G(\xi)$.

6.2 Theoretical results

There exist many results on the regularity of the constraint function and on the error between approximated solutions of chance-constrained optimization problems.

Two fundamental theorems regarding continuity and convexity of the constraint function have been proven by Raik in [57] and Prékopa in [52] in the case of multiple joint constraints. Consider problem (6.1.11) and define the function

$$\mathcal{G}(x) := \mathbb{P}\Big[G_i(x,\xi) \geq 0 \quad \forall i \in \{1,2,\dots,l\}\Big].$$

Then the continuity theorem by Raik states the following.

Theorem 6.2.1 (Raik [57])**.** *Assume ξ has an arbitrary distributed m-dimensional vector.*

- *If, for every y, the functions $G_i(\cdot, y)$ are upper semicontinuous in $\mathbb{R}^n$, then $\mathcal{G}(x)$ is upper semicontinuous in $\mathbb{R}^n$.*
- *If, for every y, the functions $G_i(\cdot, y)$ are continuous in $\mathbb{R}^n$ and*

  $$\mathbb{P}\Big[G_i(x,\xi) = 0 \quad \forall i \in \{1,2,\dots,l\}\Big] = 0$$

  then $\mathcal{G}(x)$ is continuous in $\mathbb{R}^n$.

For the convexity theorem, we first need to define the notions of *quasi-concavity* and *logconcavity*.

Definition 6.2.1 (Quasi-concave function)**.** *A function $f : \mathbb{R}^n \to \mathbb{R}$ is said to be quasi-concave if*

$$f(\lambda x + (1-\lambda)y) \geq \min\{f(x), f(y)\} \quad \forall x, y \in \mathbb{R}^n \quad \forall \lambda \in (0,1).$$

Definition 6.2.2 (Logconcave function)**.** *A function $f : \mathbb{R}^n \to \mathbb{R}$ such that $f(x) > 0$ for every $x \in \mathbb{R}^n$ is said to be logarithmically concave if*

$$f(\lambda x + (1-\lambda)y) \geq (f(x))^{\lambda}(f(y))^{1-\lambda} \quad \forall x, y \in \mathbb{R}^n \quad \forall \lambda \in (0,1)$$

$f(x) > 0$ implies that $\log f(x)$ is a concave function in $\mathbb{R}^n$.

Definition 6.2.3 (Logconcave probability measure)**.** *A probability measure defined on the Borel sets of $\mathbb{R}^n$ is said to be logarithmically concave if*

$$\mathbb{P}[\lambda A + (1-\lambda)B] \geq (\mathbb{P}[A])^{\lambda}(\mathbb{P}[B])^{1-\lambda} \quad \forall \text{ convex } A, B \subseteq \mathbb{R}^n \quad \forall \lambda \in (0,1)$$

where

$$\lambda A + (1-\lambda)B = \{z = \lambda x + (1-\lambda)y \quad | \quad x \in A, y \in B\}.$$

With these definitions, we can now state Prékopa's theorem.

Theorem 6.2.2 (Prékopa [52])**.** *If $G_i(x, y)$ is a quasi-concave function of the variables $x \in \mathbb{R}^n$ and $y \in \mathbb{R}^m$ for every i, and $\xi \in \mathbb{R}^m$ is a random variable that has a logconcave probability distribution, then the function $\mathcal{G}(x) := \mathbb{P}[G(x, \xi) \geq 0]$ is a logconcave function in $\mathbb{R}^n$.*

In [68] the authors prove that, if the random array ξ has a Gaussian distribution, it is possible to obtain a gradient formula for the nonlinear probabilistic constraint $\mathcal{G}(x)$. This is a valuable result because, even though obtaining gradient formulas in the case of linear constraints is relatively easy, doing the same in the nonlinear case is a much more difficult task. A representation formula for the gradient of $\mathcal{G}(x)$ opens the path to many solution approaches which utilize the information provided by the derivatives. Moreover, obtaining the gradient of the chance constraint is a crucial step towards establishing first order necessary conditions for optimality. Earlier results on the subject of probability functions derivatives can be found in [67] and [47].

Going back to the the convexity of the constraint function, a stronger version of Theorem 6.2.2 has been obtained by Henrion and Strugarek in the particular case where the variables x are separated from the random variables ξ:

$$\begin{cases} \min\limits_{x \in \mathcal{X}} J(x) \\ \mathbb{P}[\xi \leq G(x)] \geq p \end{cases} \tag{6.2.1}$$

where the function $G : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^m$ is vector-valued and the operator "$\leq$" represents the element-wise inequality between the m-dimensional arrays ξ and $G(x)$.

In order to state the theorem, we first need to give the definitions of *r-concave* and *r-decreasing* functions.

Definition 6.2.4 (r-concave function)**.** *A function $f : \mathbb{R}^n \to \mathbb{R}$ is said to be r-concave for some $r \in \mathbb{R}$ if*

$$f(\lambda x + (1-\lambda)y) \geq \Big(\lambda (f(x))^r + (1-\lambda)(f(y))^r\Big)^{\frac{1}{r}} \quad \forall x, y \in \mathbb{R}^n \quad \forall \lambda \in (0,1).$$

Definition 6.2.5 (r-decreasing function)**.** *A function $f : \mathbb{R} \to \mathbb{R}$ is said to be r-decreasing for some $r \in \mathbb{R}$ if it is continuous on $(0, +\infty)$ and if there exists some $x^* > 0$ such that the function $x^r f(x)$ is strictly decreasing for all $x > x^*$.*

The convexity theorem can now be stated.

Theorem 6.2.3 (Henrion and Strugarek [34])**.** *Consider problem* (6.2.1)*. Suppose that, for every i in the set $\{1, 2, \ldots, m\}$, the following hypotheses are satisfied:*

i) there exists $r_i > 0$ such that the component G_i of G is $(-r_i)$-concave;

ii) the component ξ_i of ξ is independently distributed from the others, with $(r_i + 1)$-decreasing density f_i.

Then there exists $p^* \in [0, 1]$ such that the set

$$\Big\{x \in \mathbb{R}^n \quad | \quad \mathbb{P}[\xi \leq G(x)] \geq p\Big\}$$

is convex for all $p > p^*$, where $p^* := \max_{i \in \{1,2,\ldots,m\}} F_i(x_i^*)$, F_i denotes the distribution function of ξ_i and x_i^* as in Definition 6.2.4.

Another important result in the framework of problem (6.2.1) involves the stability of the solution, and it is due to Henrion and Römisch. Let us suppose that we only have partial information on the probability measure μ and that we replace it with some estimation $\nu \in \mathcal{P}(\mathbb{R}^m)$. We then define the following functions

$$\psi(\nu) := \arg\min_{x \in \mathcal{X}} \Big\{J(x) \quad | \quad F_\nu(G(x)) \geq p\Big\}$$
$$\phi(\nu) := \inf_{x \in \mathcal{X}} \Big\{J(x) \quad | \quad F_\nu(G(x)) \geq p\Big\}$$

where $F_\nu(y) := \nu(z \in \mathbb{R}^m \quad | \quad z \leq y)$ is the distribution function of the probability measure ν. We point out that, in general, the solutions of (6.2.1)

are not unique under the assumptions that will be later stated in Theorem 6.2.4. Because of this, in the definition of ψ and ϕ we have to deal with solution sets. The distances among parameters and solutions are measured respectively with the Kolmogorov and Hausdorff metrics:

$$d_K(\nu_1, \nu_2) := \sup_{y \in \mathbb{R}^m} \left| F_{\nu_1}(y) - F_{\nu_1}(y) \right| \quad \nu_1, \nu_2 \in \mathcal{P}(\mathbb{R}^m)$$
$$d_H(A, B) := \max \left\{ \sup_{a \in A} d(a, B), \sup_{b \in B} d(b, A) \right\} \quad A, B \subseteq \mathbb{R}^m.$$

Before stating the theorem we have to extend the definition of r-concavity to a probability measure.

Definition 6.2.6 (r-concavity)**.** *The probability measure μ is said to be r-concave if μ^r is a convex set function, i.e. for any $\lambda \in (0,1)$ and all Borel measurable and convex sets $A, B \in \mathbb{R}^m$ such that $\lambda A + (1-\lambda)B$ is Borel measurable the following inequality holds*

$$\mu^r(\lambda A + (1-\lambda)B) \leq \lambda \mu^r(A) + (1-\lambda)\mu^r(B).$$

Moreover, we define

$$Y_V := \left[G(\mathcal{X} \cap \overline{V}) + \mathbb{R}^m_- \right] \cap F_\mu^{-1} \left(\left[\frac{p}{2}, 1 \right] \right)$$
$$\pi(y) := \inf_{x \in \mathcal{X} \cap \overline{V}} \{ J(x) \quad | \quad G(x) \geq y \}$$
$$\sigma(y) := \arg\min_{x \in \mathcal{X} \cap \overline{V}} \{ J(x) \quad | \quad G(x) \geq y \} \quad y \in Y_V$$
$$Y(\nu) := \arg\min_{y \in Y_V} \{ \pi(y) \quad | \quad F_\nu(y) \geq p \} \quad \nu \in \mathcal{P}(\mathbb{R}^m).$$

Theorem 6.2.4 (Henrion and Römisch [33])**.** *Assume that $\mathcal{X}$ is closed, J and $\mathcal{X}$ are convex, G has concave components and the following holds:*

- *μ is r-concave;*
- *$\psi(\mu)$ is nonempty and bounded;*
- *there exists some $\hat{x} \in \mathcal{X}$ such that $F_\mu(G(\hat{x})) > p$;*
- *F_μ^r is strongly convex on some convex open neighborhood U of $Y(\mu)$, where $r < 0$ is chosen such that μ is r-concave;*
- *σ is Hausdorff Hölder continuous with rate κ^{-1} on Y_V.*

Then ψ is Hausdorff Hölder continuous with rate $(2\kappa)^{-1}$ at μ, i.e. there are $L > 0$ and $\delta > 0$ such that

$$d_H(\psi(\mu), \psi(\nu)) \leq L(d_K(\mu, \nu))^{\frac{1}{2\kappa}} \quad \forall \nu \in \mathcal{P}(\mathbb{R}^m) \quad | \quad d_K(\mu, \nu) < \delta.$$

Chapter 7

Approximation of chance-constrained problems

7.1 Numerical approaches

The most intuitive methods for solving a problem in the form of (6.1.10) belong to the class of Monte Carlo methods. An algorithm of this type consists in repeatedly sampling variables and parameters of a problem in order to obtain numerical results, treating them as random quantities. This kind of approach might be very useful in the case of problems involving a high number of dimensions, many degrees of freedom or unknown probability distributions.

The first modern definition of the Monte Carlo method was developed in the late '40s by Stan Ulam. Shortly after, John von Neumann understood the importance of the technique and implemented it for the first electronic computer: the ENIAC [48]. During the following years, the popularity of the Monte Carlo method grew in parallel with the increasing power of computers.

The general procedure of a method belonging to the Monte Carlo class consists in performing the following steps:

1. Define the inputs of the problem as well as their domain.
2. Choose a probability distribution for the inputs and generate random input values over the domain.
3. Elaborate the results using a deterministic mathematical model.

The mathematical theory supporting these methods depends on the particular type chosen, but the main result on which the whole Monte Carlo methods' class lays foundation is the Strong Law of Large Numbers.

Theorem 7.1.1 (Strong Law of Large Numbers)**.** *Let $X_1, X_2, \ldots, X_n$ be a sequence of independent identically distributed random variables, each one*

with finite average $\mu = \mathbb{E}[X_i]$ *for every* $i \in \{1, 2, \dots, n\}$*, then*

$$\mathbb{P}\left[\lim_{n\to+\infty} \frac{\sum_{i=1}^{n} X_i}{n} = \mu\right] = 1.$$

A simple proof for this result can be found in [58]. This theorem can be applied for estimating the probability of a random variable via a sequence of samples, and we will use this application later on for the verification of the numerical tests. Let E be a given event, relative to a single realization of a random variable X. By choosing a number $n \in \mathbb{N}$ of tries and defining for every $i \in \{1, 2, \dots, n\}$

$$X_i = \begin{cases} 1 & E \text{ realizes at the } i\text{-th try} \\ 0 & \text{else} \end{cases}$$

we can apply Theorem 7.1 to obtain

$$\mathbb{P}\left[\lim_{n\to+\infty} \frac{\sum_{i=1}^{n} X_i}{n} = \mathbb{E}[X] = \mathbb{P}[E]\right] = 1.$$

There are many advantages to Monte Carlo methods: they are usually easy to implement and can be parallelized if the random variables to be sampled are independent. Moreover, given the wide variety of existing Monte Carlo methods, it is not difficult to find an implementation specifically designed for a particular field: from physics to statistics, from biology to finance, as well as engineering and Artificial Intelligence.

The rest of this section is dedicated to the description of two techniques for the numerical solution of chance-constrained optimization problems which take inspiration from the Monte Carlo-type approach: The Stochastic Arrow-Hurwicz Algorithm and the Kernel Density Estimation.

7.1.1 Stochastic Arrow-Hurwicz Algorithm

The stochastic Stochastic Arrow-Hurwicz Algorithm (SAHA) is designed to solve an optimization problem in the form

$$\begin{cases} \min_{x\in\mathcal{X}} \mathbb{E}[K(x,\xi)] \\ \mathbb{E}[H(x,\xi)] \leq \alpha. \end{cases} \tag{7.1.1}$$

The constraint in expectation can be rewritten explicitly as a chance constraint by defining the function G as

$$H(x,\xi) := -\mathbf{1}_{\mathbb{R}^+}(G(x,\xi)) = \begin{cases} -1 & G(x,\xi) \geq 0 \\ 0 & \text{else} \end{cases}$$

to obtain

$$\mathbb{E}[H(x,\xi)] = -\mathbb{P}[G(x,\xi) \geq 0]$$

then, by setting $\alpha := -p$ we have

$$\mathbb{E}[H(x,\xi)] \leq \alpha \Leftrightarrow \mathbb{P}[G(x,\xi) \geq 0] \geq p.$$

To cite a successful implementation of the SAHA in aerospace, in [18] the authors solve the problem of minimizing fuel consumption while driving a satellite to its final position with a certain probability threshold, considering that the engine may fail at a random instant for a random amount of time.

This algorithm arises from the combination of the Monte Carlo method's idea with the iterative gradient method. In the case of a deterministic constrained optimization problem in the form

$$\begin{cases} \min\limits_{x\in\mathcal{X}} K(x) \\ H(x) \leq \alpha \end{cases}$$

if there exists a saddle point for the Lagrangian

$$L(x,\lambda) = K(x) + \langle \lambda, H(x) - \alpha \rangle$$

the deterministic Arrow-Hurwicz Algorithm consists in iterating the following steps: let x_k and λ_k be the estimates for the primal and dual variable at iteration k, then define

$$\begin{aligned} x_{k+1} &:= \Pi_{\mathcal{X}}\Big(x_k - \varepsilon_k\big(\nabla_x K(x_k) + \lambda_k \nabla_x H(x_k)\big)\Big) \\ \lambda_{k+1} &:= \Pi_{\mathbb{R}_+}\Big(\lambda_k + \rho_k\big(H(x_{k+1}) - \alpha\big)\Big) \end{aligned}$$

where the sequences $\{\varepsilon_k\}$ and $\{\rho_k\}$ affect the length of the each approximation step and Π_A denotes the projection onto the set A.

With the introduction of the random variable ξ the stochastic version of this algorithm takes the form

1. draw ξ_{k+1} according to the distribution law of ξ
2. update x_k and λ_k:

$$x_{k+1} := \Pi_{\mathcal{X}}\Big(x_k - \varepsilon_k\big(\nabla_x K(x_k,\xi_{k+1}) + \lambda_k \nabla_x H(x_k,\xi_{k+1})\big)\Big) \tag{7.1.2}$$

$$\lambda_{k+1} := \Pi_{\mathbb{R}^m_+}\Big(\lambda_k + \rho_k\big(H(x_{k+1},\xi_{k+1}) - \alpha\big)\Big). \tag{7.1.3}$$

Culioli and Cohen proved the following convergence theorem in the general case where the functions K and H might not be differentiable in the classic sense.

Theorem 7.1.2 (Culioli and Cohen [22])**.** *In addition to general measurability assumptions, we suppose that the associated lagrangian L admits a saddle point $(\tilde{x},\tilde{\lambda})$, and that*

i) $\mathbb{E}[K(x,\xi)]$ *is strictly convex on* $\mathcal{X}$ *and, for every* $\xi \in \Omega$ *,*$K(x,\xi)$ *is locally Lipschitz on* $\mathcal{X}$*.*

ii) For all $\xi \in \Omega$*,* $H(x,\xi)$ *is locally Lipschitz on* $\mathcal{X}$ *and regularly subdifferentiable, with subgradient with respect to* x *uniformly bounded. We also assume that it is sub-Lipschitz, i.e. for every* $\xi \in \Omega$

$$\|H(x,\xi) - H(y,\xi)\| \leq L_H\|x-y\| + \mu_H \quad \forall x, y \in \mathcal{X}$$

and that $\mathbb{E}[H(x,\xi)]$ *is* C*-convex and Lipschitz.*

iii) There exist positive constants α_K *and* β_K *such that, for every* $\xi \in \Omega$ *and* $r \in \partial_x K(x,\xi)$

$$\|r\| \leq \alpha_K\|x-\tilde{x}\| + \beta_K \quad \forall x \in \mathcal{X}$$

where $\partial_x K$ *denotes the Clarke subdifferential of* K *with respect to* x*.*

iv) The sequences $\{\varepsilon_n\}_{n\in\mathbb{N}}$ *and* $\{\rho_n\}_{n\in\mathbb{N}}$ *are* σ*-sequences, i.e.*

$$\sum_{n=0}^{+\infty} \varepsilon_n = +\infty \quad \textit{and} \quad \sum_{n=0}^{+\infty} (\varepsilon_n)^2 < +\infty$$

$$\sum_{n=0}^{+\infty} \rho_n = +\infty \quad \textit{and} \quad \sum_{n=0}^{+\infty} (\rho_n)^2 < +\infty$$

and the sequence $\left\{\frac{\varepsilon_n}{\rho_n}\right\}_{n\in\mathbb{N}}$ *is monotone.*

v) there exist positive constants γ_H *and* δ_H *such that*

$$\mathbb{E}\left[\Big(H(x,\xi) - \mathbb{E}[H(x,\xi)]\Big)^2\right] < \gamma_H\|x-\tilde{x}\|^2 + \delta_H \quad \forall x \in \mathcal{X}.$$

Then, almost surely, the sequence $\{(x_n,\lambda_n)\}_{n\in\mathbb{N}}$ *is bounded and* $\{x_n\}_{n\in\mathbb{N}}$ *weakly converges to* $\tilde{x}$*. Moreover, if* $\mathbb{E}[K(x,\xi)]$ *is strongly convex, then the sequence* $\{x_n\}_{n\in\mathbb{N}}$ *strongly converges to* $\tilde{x}$*.*

As for the convergence rate, the general result in [41, Theorem 3.1] can be applied to the case of the SAHA to obtain an upper bound for the Asymptotic Mean Squared Error (AMSE). In order to state the result, we first need to rewrite algorithm (7.1.2) in the following compact form

$$z_{k+1} = \Pi(z_k - w_k \cdot \psi(z_k, \xi_{k+1})^\intercal)$$

where $z_k := (x_k, \lambda_k)$, $w_k := (\varepsilon_k, \rho_k)$,

$$\psi(z_k,\xi_{k+1}) := (\nabla_x K(x_k,\xi_{k+1}) + \lambda_k \nabla_x H(x_k,\xi_{k+1}), H(x_{k+1},\xi_{k+1}) - \alpha)$$

and Π stands for the projection operation on $\mathcal{X} \times \mathbb{R}^m_+$. Then we choose $\gamma > 0$, set $\varepsilon_k = \frac{1}{k^\gamma}$, ρ_k proportional to ε_k and define

$$\begin{aligned}
\Psi(\tilde{z}) &:= \Big(\nabla_x \mathbb{E}[K(x,\xi)] + \lambda \nabla_x \mathbb{E}[H(x,\xi)], \mathbb{E}[H(x,\xi)] - \alpha\Big) \\
B_k &:= \mathbb{E}[\psi(z_k,\xi)] - \Psi(z_k) \\
V_k &:= \mathbb{E}\left[\left\|\psi(z_k,\xi) - \mathbb{E}[\psi(z_k,\xi)]\right\|^2\right].
\end{aligned}$$

Finally, we choose $\beta > 0$ and $\delta > 0$ such that $B_k = O\left(\frac{1}{k^\beta}\right)$ and $V_k = O\left(\frac{1}{k^\delta}\right)$, and obtain

$$\mathrm{AMSE}(z_k)\mathbb{E}[(z_k - \tilde{z})^2] \leq O(k^{-\min\{2\beta, \gamma+\delta\}}).$$

This result shows that the higher $\min\{2\beta, \gamma+\delta\}$ is, the faster the sequence $\{z_k\}_{n\in\mathbb{N}}$ converges to $\tilde{z} := (\tilde{x}, \tilde{\lambda})$.

Unfortunately this algorithm has also some disadvantages. In the particular case of chance-constrained optimization problems, the convexity and the connectedness of the feasible subset defined by H are not easily satisfied, this implies that the existence of a saddle point of the Lagrangian is not granted. The reformulation of the probability constraint as a constraint in expectation is not harmless either: defining H by means of a characteristic function introduces a discontinuity that must be carefully regularized in order to evaluate $\nabla_x H$. See [2] for a more detailed explanation of the above-mentioned difficulties.

7.1.2 Kernel Density Estimation

Another approach for solving the chance-constrained problem numerically is based on Kernel Density Estimation, a particular kind of density estimation used in statistics. This technique consists in approximating the probability density function (pdf) of a random variable with unknown distribution from a given sample. This technique has also been applied to many other fields like archaeology, banking, climatology, economics, genetics, hydrology and physiology (see [65] for more references). Silverman's book [66] represents the basic text on the subject.

In the case of problem (6.1.10), if we are able to produce an approximation of the pdf defining the chance constraint, we can replace the probability with the integral of the estimated pdf and solve the stochastic optimization problem as a deterministic one. For a given x in $\mathcal{X}$, let f_x and $\hat{f}_x$ be respectively the pdf of $G(x,\xi)$ and its approximation. From basic probability theory, we have

$$\mathbb{P}[G(x,\xi) \geq 0] = 1 - \mathbb{P}[G(x,\xi) < 0] = 1 - \int_{-\infty}^{0} f_x(z)dz$$

if f_x and $\hat{f}_x$ are "close" in some appropriate sense, we obtain

$$\int_{-\infty}^{0} \hat{f}_x(z)dz \approx \int_{-\infty}^{0} f_x(z)dz = 1 - \mathbb{P}[G(x,\xi) \geq 0].$$

By defining $\hat{F}_x(y) := \int_{-\infty}^{y} \hat{f}_x(z)dz$ we can write an approximation of problem (6.1.10) in the form

$$\begin{cases} \min\limits_{x\in\mathcal{X}} J(x) \\ \hat{F}_x(0) \leq 1-p. \end{cases} \tag{7.1.4}$$

Let x^* and $\hat{x}^*$ be respectively the solutions of problems (6.1.10) and (7.1.4), if we can control the error between x^* and $\hat{x}^*$ by means of the error between $\hat{f}_x$ and f_x, we could be able to produce an approximate solution to our original problem which can be as accurate as we need. Let X be a random variable with an unknown distribution f that we want to estimate and let $\{X_1, X_2, \ldots, X_n\}$ be a sample of size n from the variable X. A Kernel Density Estimator for the pdf f is the function

$$\hat{f}_{n,h}(x) := \frac{1}{nh}\sum_{i=1}^{n} K\left(\frac{x - X_i}{h}\right) \tag{7.1.5}$$

where the function K is called *kernel* and the smoothing parameter h is called *bandwidth*. A fundamental result has been obtained by Nadaraya:

Theorem 7.1.3 (Nadaraya [49])**.** *If the kernel $K : \mathbb{R} \to \mathbb{R}_+$ is a function of bounded variation, $f : \mathbb{R} \to \mathbb{R}_+$ is a uniformly continuous density function, and if h satisfies*

$$\sum_{n=1}^{+\infty} e^{-\gamma n h^2} < +\infty \quad \forall \gamma > 0$$

then

$$\mathbb{P}\Big[\lim_{n\to+\infty} \sup_x |\hat{f}_{n,h}(x) - f(x)| = 0\Big] = 1.$$

The approximation error between f and $\hat{f}_{n,h}$ depends on the choice of both K and h. The kernel K is generally chosen such that it satisfies the conditions

$$\int K(y)dy = 1 \quad \text{and} \quad \int yK(y)dy = 0 \quad \text{and} \quad \int y^2K(y)dy > 0$$

in most applications the study is focused on the choice of h and K is usually the Gaussian kernel:

$$K(x) := \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}}.$$

If the unknown density is sufficiently smooth and the kernel has a finite fourth moment (which is true for of the Gaussian kernel) we can use Taylor

expansions to show that

$$\begin{aligned}\text{Bias}[\hat{f}_{n,h}(x)] &= \frac{h^2}{2}\int y^2 K(y)dy f''(x) + o(h^2)\\ \text{Var}[\hat{f}_{n,h}(x)] &= \frac{1}{nh}\int K^2(y)dy f(x) + o\left(\frac{1}{nh}\right)\end{aligned}$$

where $\text{Bias}[\hat{f}(x)]$ and $\text{Var}[\hat{f}(x)]$ denote the estimator bias and variance respectively. With these definitions we can use the Mean Integrated Squared Error (MISE) as a measure of the discrepancy between $\hat{f}$ and f:

$$\begin{aligned}\text{MISE}[\hat{f}(x)] &:= \mathbb{E}\left[\int \left(\hat{f}_{n,h}(x) - f(x)\right)^2 dx\right] =\\ &= \int \text{Bias}^2[\hat{f}_{n,h}(y)]dy + \int \text{Var}[\hat{f}_{n,h}(y)]dy.\end{aligned}$$

Under integrability assumptions on f (see [65]), the MISE becomes the Asymptotic Mean Integrated Squared Error (AMISE):

$$\text{AMISE}[\hat{f}_{n,h}(x)] := \frac{1}{nh}\int K^2(y)dy + \frac{h^4}{4}\left(\int y^2 K(y)dy\right)^2 \int f''^2(y)dy$$

which leads to the following optimal choice for the bandwidth h

$$h_{\text{AMISE}} := \sqrt[5]{\frac{\int K^2(y)dy}{n\left(\int y^2 K(y)dy\right)^2 \int f''^2(y)dy}}. \tag{7.1.6}$$

Unfortunately, the presence of the unknown factor $\int f''^2(y)dy$ in the previous definition makes the expression of h_{AMISE} almost useless. For this reason there exist many practical ways (see [65]) for choosing the bandwidth using only informations on the sample. A common choice for h, used in conjunction with the Gaussian kernel, is the Simple Normal Reference (SNR): let S be the sample standard deviation, the SNR bandwidth is then defined as

$$h_{\text{SNR}} := 1.06\frac{S}{\sqrt[5]{n}}. \tag{7.1.7}$$

Even though there's no general rule for obtaining an explicit value of h leading to the best approximation of f, it's important to point out that big values of h will probably lead to an overestimation of the volume of the density function and thus to a loss of information.

7.2 Numerical results

This section is dedicated to some numerical applications of the Stochastic Arrow-Hurwicz Algorithm and the Kernel Density Estimation. Before showing the results we will explain how they are obtained from a coding point of view, describing the solvers used and their interface with programming languages.

7.2.1 Nonlinear optimization solvers

Most of the time, whenever we need to numerically compute a solution to an optimization problem, we have to write a code interface using a specific programming or scripting language (e.g. C, C++, Fortran, MATLAB, AMPL,...). The purpose of this interface is to translate our problem's abstract formulation into one accepted by one of the available solvers for nonlinear optimization (e.g. IPOPT, WORHP, KNITRO,...).

The results in this section have been obtained using Fortran 90 to write the code interface and both WORHP and IPOPT as solvers. These two solvers are designed to handle finite dimensional nonlinear optimization problem in the form

$$\begin{cases} \min\limits_{X\in\mathbb{R}^N} F(X) \\ X_{\mathrm{L}} \leq X \leq X_{\mathrm{U}} \\ G_{\mathrm{L}} \leq G(X) \leq G_{\mathrm{U}} \end{cases} \tag{7.2.1}$$

where $N \in \mathbb{N}$ is the number of decision (or optimization) variables, which are collected in the array $X := (X_1, X_2, \ldots, X_N)$; the function $F(X) : \mathbb{R}^N \to \mathbb{R}$ represents the cost to be minimized; $G(X) : \mathbb{R}^N \to \mathbb{R}^M$ is the constraint function, with $M \in \mathbb{N}$ being the number of constraints to be satisfied. The arrays $X_{\mathrm{L}}, X_{\mathrm{U}} \in \mathbb{R}^N$ and $G_{\mathrm{L}}, G_{\mathrm{U}} \in \mathbb{R}^M$ define respectively the lower and upper bounds for X and G. In addition to this, the user must provide an initial guess X_0 for the solution of (7.2.1), while the derivatives of F and G are optional since they can be usually approximated by the solvers themselves.

WORHP (We Optimize Really Huge Problems) implements a Sequential Quadratic Programming (SQP) method which is based on a descent method with line search. For more details on this algorithm, consult the *User's Guide to WORHP* available at `www.worhp.de`.

IPOPT (Interior Point OPTimizer) instead, implements, as the name suggests, an interior point line search filter method. We refer to *Introduction to IPOPT: A tutorial for downloading, installing, and using IPOPT* (available at `https://projects.coin-or.org/Ipopt`) for a step-by-step guide for interfacing the solver with several programming languages, and to [69] for a detailed analysis of the mathematical background.

7.2.2 Test 1: Simple single stage launcher with one decision variable and one random variable

The first numerical test involves a very simple chance constrained optimization problem in the domain of aerospace engineering. Consider the vertical ascent of a single stage launcher (i.e. consisting in one continuous structure with no detachable parts) in one dimension. At a given time t, we only measure the position of the launcher by means of its altitude $r(t)$ and represent

its speed with a scalar value $v(t)$. The vehicle starts from a still position at ground level, and at time $t = 0$ the thrust force T of the engine pushes the launcher upwards against the force of gravity $m(t)g$, where $m(t)$ denotes the mass of the launcher at time t. The rate at witch the mass is consumed is the ratio between thrust force T and the fuel speed v_e (which are assumed to be constant over time).

We can act on the dynamics only by changing the initial mass m_0, meaning that we only have one decision variable. We are interested in studying the case where the parameter T is a random variable, i.e. the value of T is unknown at $t = 0$ and it changes randomly for each launch. Our goal is then to minimize the initial mass of the launcher while guaranteeing some constraints to be satisfied even if T is subject to random variations: an equality constraint on the launcher's final position (its final position has to match *exactly* the value we need) and a chance constraint on its final mass (the final mass has to surpass a given threshold $\overline{m}_u$ with a probability of *at least* p).

A solution to this problem consists in an optimal value m_0^* which is the smallest amount of mass that allows the launcher to satisfy the chance constraint. This means that if (for a given realization of T) we define a launch to be successful when the final mass is higher than $\overline{m}_u$, we expect that over a large number of launches with initial mass m_0^*, the ratio between successful launches and total attempts is close to p. In this case m_0^* can be considered a "relaxed" robust solution to our constrained optimization problem, in the sense that it works in presence of variations in the parameters defining the problem while still allowing some margin of error.

Model

The ODE system describing the dynamics is

$$\begin{cases} \dot{r}(t) = v(t) \\ \dot{v}(t) = \frac{T}{m(t)} - g \\ \dot{m}(t) = -\frac{T}{v_e} \\ r(0) = 0 \\ v(0) = 0 \\ m(0) = m_0 < \frac{T}{g}. \end{cases}$$

The last inequality makes sure that the launcher his not too heavy or, equivalently, that the engine is powerful enough to overcome the force of gravity.

The system can be solved explicitly:

$$\begin{aligned} r(t) &= \left(v_{\mathrm{e}}t - \frac{v_{\mathrm{e}}^2 m(0)}{T}\right)\ln\left(\frac{m(0)}{m(t)}\right) + v_{\mathrm{e}}t - \frac{g}{2}t^2 \\ v(t) &= v_{\mathrm{e}}\ln\left(\frac{m(0)}{m(t)}\right) - gt \\ m(t) &= m_0 - \frac{T}{v_{\mathrm{e}}}t. \end{aligned}$$

Before defining our stochastic optimization problem, let us first look at the solution to its deterministic counterpart:

$$\begin{cases} \min\limits_{m_0\in\mathbb{R}} m_0 \\ M_{\mathrm{u}}(m_0) \geq \overline{m}_{\mathrm{u}}. \end{cases} \tag{7.2.2}$$

The constraint function is defined as

$$M_{\mathrm{u}}(m_0) := m_0 - (1+k)m_{\mathrm{e}}(m_0).$$

In the previous formula, the parameter k is called *stage index* and it represents the ratio between the structure's mass and the fuel mass $m_{\mathrm{e}}(m_0) := \frac{T}{v_{\mathrm{e}}}t_{\mathrm{f}}(m_0)$. For a given m_0, the final time $t_{\mathrm{f}}(m_0)$ is the smallest time that satisfies the energy constraint on the final orbit:

$$t_{\mathrm{f}}(m_0) := \min\left\{t \in [0,+\infty) \text{ s.t. } r(t) + \frac{v^2(t)}{2g} = \omega_{\mathrm{f}}\right\}.$$

Note that, being $r(t) + \frac{v^2(t)}{2g}$ an increasing monotone function of time (see figure 7.2.1, bottom right), the previous formula can be reduced to the solution of the equation $r(t) + \frac{v^2(t)}{2g} = \omega_{\mathrm{f}}$. Since this equation cannot be solved explicitly, its solution will be computed numerically with a standard bisection method. Using the values in Table 7.2.1, we compute the set of

Parameter	T	k	v_{e}	g	ω_{f}	$\overline{m}_{\mathrm{u}}$
Value	150 [N]	0.1	5 [m/s]	9.8 [m/s^2]	0.5 [m]	0.5 [kg]

Table 7.2.1: Parameters for the deterministic optimization

values for m_0 satisfying the inequalities $0 < m_0 < \frac{T}{g}$ and obtain that the interval of admissible values for the initial mass is $(0, \approx 15,3001)$.

We can solve this problem using a nonlinear optimization solver, as long as we rewrite it in the form (7.2.1). The optimal solution found by WORHP is

$$m_0^* \approx 1.04709 \text{ [kg]}.$$

Figures 7.2.1 show the corresponding optimal trajectory.

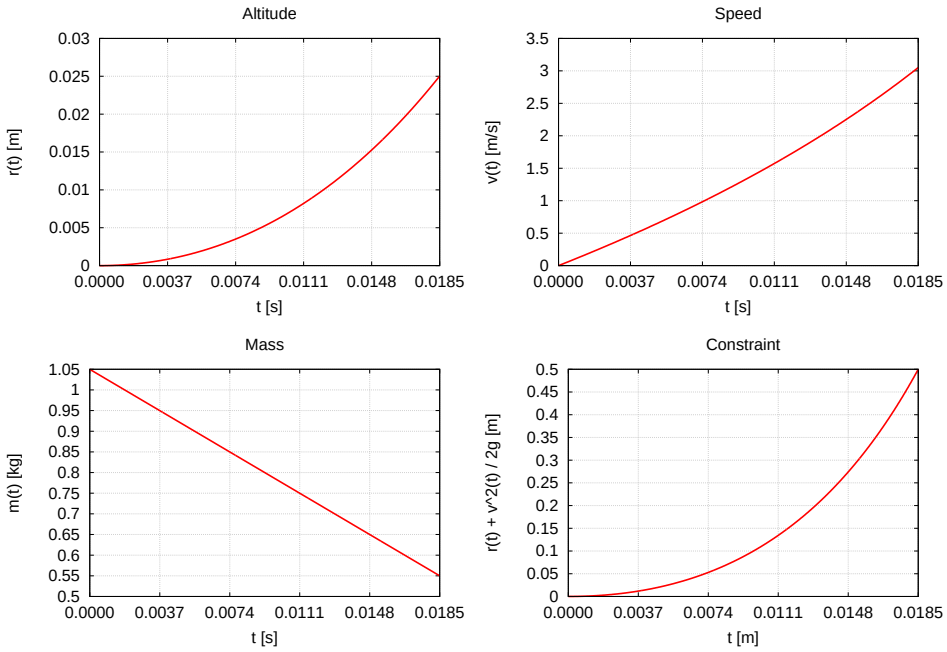


Figure 7.2.1: Plot of altitude, speed, mass and constraint for the single-stage launcher.

Problem statement

Since we want to deliver a given payload $\overline{m}_{\mathrm{u}}$ with a 90% probability while minimizing the initial total mass of the launcher (and thus the fuel mass), we define the stochastic optimization problem

$$\begin{cases} \min\limits_{m_0 \in \mathbb{R}} m_0 \\ \mathbb{P}\big[M_{\mathrm{u}}(T, m_0) \geq \overline{m}_{\mathrm{u}}\big] \geq p \end{cases} \tag{7.2.3}$$

where T is a uniformly distributed random variable on the interval $[T_-, T_+]$ with expected value $\overline{T}$:

$$T \sim U(T_-, T_+).$$

Here $T_- := \overline{T}(1 - \Delta T)$, $T_+ := \overline{T}(1 + \Delta T)$. The pdf (probability density function) of T is

$$\phi(x) := \begin{cases} \frac{1}{T_+ - T_-} & x \in [T_-, T_+] \\ 0 & \text{else.} \end{cases}$$

$M_{\mathrm{u}}(T, m_0)$ is a function of the random variable T, parameterized by m_0:

$$M_{\mathrm{u}}(T, m_0) := m_0 - (1 + k) m_{\mathrm{e}}(T, m_0).$$

Table 7.2.2 shows the choice of parameters defined in this subsection. Figure

Parameter	p	$\overline{T}$	ΔT
Value	0.9	150 [N]	0.1

Table 7.2.2: Additional parameters for the stochastic optimization

7.2.2 shows the plot of $M_{\mathrm{u}}(T, m_0)$ as a function of m_0 for $T = \overline{T}$. We can observe that there is a critical value for the initial mass m_0 (≈ 15.3) above which it's impossible to satisfy the constraint on the final altitude because the launcher is too heavy (therefore $t_{\mathrm{f}}(m_0) = +\infty$), implying that the payload delivered is zero. It is also interesting to remark how the payload mass is not a monotone increasing function of m_0 but it actually has a maximum.

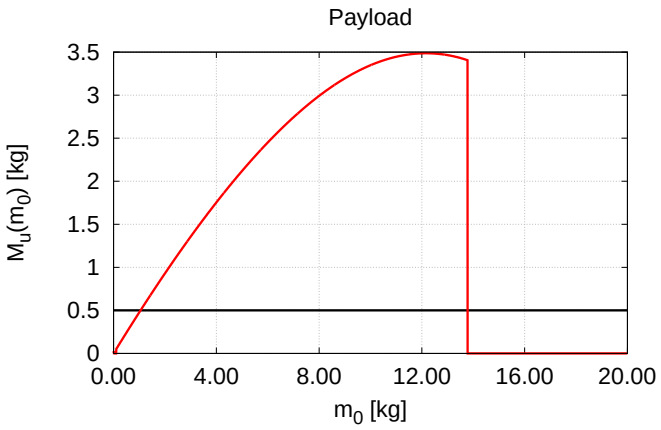


Figure 7.2.2: Plot of the payload as a function of m_0.

Solution via Stochastic Arrow-Hurwicz Algorithm

In order to apply the SAHA we have to rewrite problem (7.2.3) in the form (7.1.1). The only decision variable is $x := m_0$, while $\mathcal{X} := \mathbb{R}^+$ is the set of admissible solutions. The cost functional then becomes $F(x, \xi) := m_0 = x$, since it doesn't depend on $\xi := T$, the only random variable. We also define the function G as

$$G(x, \xi) := -\mathbf{1}_{\mathbb{R}^+}\big(M_{\mathrm{u}}(\xi, x) - \overline{m}_{\mathrm{u}}\big) = \begin{cases} -1 & M_{\mathrm{u}}(\xi, x) \geq \overline{m}_{\mathrm{u}} \\ 0 & \text{else} \end{cases}$$

and we obtain

$$\mathbb{E}\big[G(x, \xi)\big] \leq \alpha \Leftrightarrow \mathbb{P}\big[M_{\mathrm{u}}(\xi, x) \geq \overline{m}_{\mathrm{u}}\big] \geq p$$

where $\alpha := -p$. Following [18], the sequences $\{\varepsilon_k\}$ and $\{\rho_k\}$ are defined as

$$\varepsilon_k := \frac{a_\varepsilon}{b_\varepsilon + k} \quad \rho_k := \frac{a_\rho}{b_\rho + k}$$

and also set $a_\rho := 2a_\varepsilon$ and $b_\rho := b_\varepsilon$ to ensure that hypotheses *iv)* of Theorem 7.1.2 are satisfied. Along the same lines of [18], the indicator function defining the constraint in expectation has been replaced by a smoother version which depends on a parameter r:

$$\mathbf{1}_{\mathbb{R}^+}(x) \approx I_r(x) := \begin{cases} 1 & x \geq 0 \\ \left(1 - \left(\frac{x}{r}\right)^2\right)^2 & -r \leq x \leq 0 \\ 0 & \text{else.} \end{cases}$$

In our case, we choose r to be dependent on k, so that as k increases, r_k tends to zero:

$$r_k := \frac{a_r}{b_r + k}.$$

It should be noted that there are many other ways to approximate the expectation of an indicator function. A more detailed analysis of this approach can be found in [50, 29].

Results

We solved the problem with different initializations for x_0, which is the initial guess for the solution. The optimal solution m_0^* is defined as the last value of the sequence $\{x_k\}$ after 10^5 iterations. Table 7.2.3 summarizes the choice of parameters for the algorithm. The tuning of these parameters is quite difficult and unfortunately it has to be done heuristically since it depends heavily on the particular formulation of the problem and there's no general rule for choosing them.

Parameter	λ_0	a_ε	b_ε	a_ρ	b_ρ	a_r	b_r
Value	0.005	0.5	1000	1	1000	100	1000

Table 7.2.3: Parameters for the SAHA

We will check each solution by applying Theorem 7.1: for each n, we will call m_0^* the optimal solution found and then draw a large random sample N_a from T. Let N_s be the number of times that the event $M_\mathrm{u}(T, m_0^*) \geq \overline{m}_\mathrm{u}$ occurs. The Strong Law of Large Numbers states that, with probability 1

$$\lim_{N_\mathrm{s} \to +\infty} \frac{N_\mathrm{s}}{N_\mathrm{a}} = \mathbb{P}[M_\mathrm{u}(T, m_0^*) \geq \overline{m}_\mathrm{u}].$$

Table 7.2.4 and Figure 7.2.3 show the comparison between the five different cases.

x_0	m_0^* [kg]	R
4	**1.6926921752**	**1.0000**
3.5	**1.1926921752**	**1.0000**
3	**1.0495447154**	**0.9506**
2.5	**1.0486332974**	**0.7931**
2	**1.0486458316**	**0.7955**
1.5	**1.0486458316**	**0.7955**
1	**0.0000000000**	**0.0000**
0.5	**0.0000000000**	**0.0000**

Table 7.2.4: Optimal solution m_0^* and success rate R for each choice of x_0.

Solution via Kernel Density Estimation

In order to use the KDE we have to reformulate the chance constraint showing its dependency on the CDF F of random variable M_u. Let f_{m_0} be the pdf of M_u, parameterized by m_0. From the definition of f_{m_0} we have

$$\mathbb{P}\big[M_\mathrm{u}(T, m_0) \geq \overline{m}_\mathrm{u}\big] = 1 - \mathbb{P}\big[M_\mathrm{u}(T, m_0) < \overline{m}_\mathrm{u}\big] = 1 - \int_0^{\overline{m}_\mathrm{u}} f_{m_0}(x)dx.$$

Then, if we define

$$F_{m_0}(\overline{m}_\mathrm{u}) := \int_0^{\overline{m}_\mathrm{u}} f_{m_0}(x)dx$$

we can rewrite problem 3.1.1 as

$$\begin{cases} \min\limits_{m_0 \in \mathbb{R}} m_0 \\ F_{m_0}(\overline{m}_\mathrm{u}) \leq 1 - p. \end{cases} \tag{7.2.4}$$

For each value of m_0 we are able to produce an approximation $\hat{F}_{m_0}$ of F_{m_0} via KDE by drawing a sample of size n from the random variable T. Our problem then becomes

$$\begin{cases} \min\limits_{m_0 \in \mathbb{R}} m_0 \\ \hat{F}_{m_0}(\overline{m}_\mathrm{u}) \leq 1 - p. \end{cases} \tag{7.2.5}$$

The procedure used for solving problem (7.2.5) is described in the following steps.

1. **Draw the sample**
   We can either take n equidistant values for T in the interval $[T_-, T_+]$ such that
   $$T_i = T_- + (i-1)\frac{T_+ - T_-}{n-1} \quad \forall i \in 1, \ldots, n$$

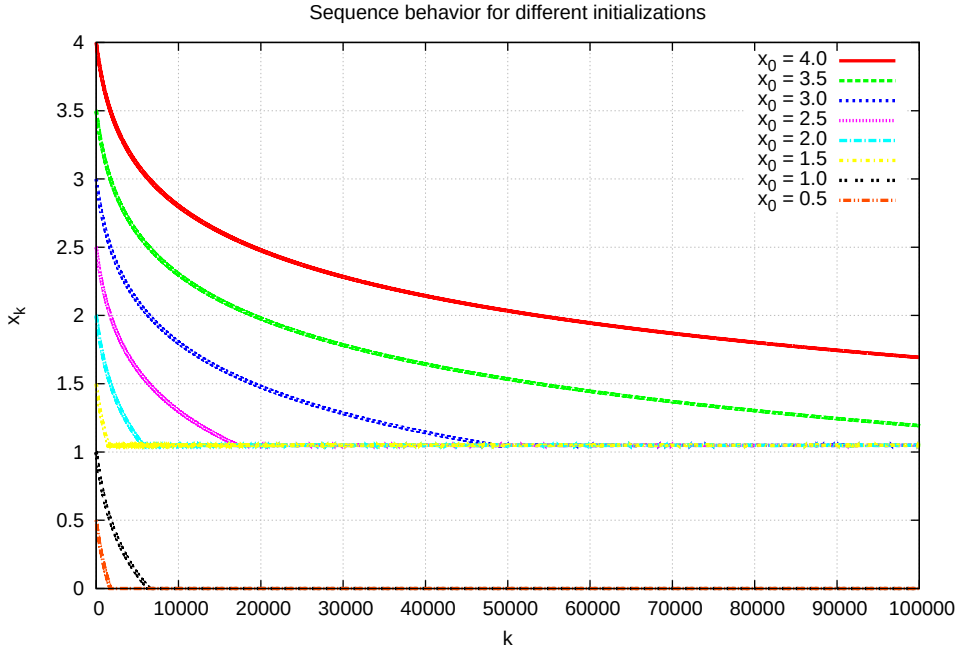


Figure 7.2.3: Plot of the sequence x_k for each choice of x_0.

or draw n random realizations from the variable T according to its distribution law. While the first approach clearly leads to more accurate results, it may be computationally demanding when the number of random parameters grows.

2. **Define the constraint function**
   Instruct the solver on how to associate an initial mass m_0 to the constraint function $\hat{F}_{m_0}$.

   (a) For each T_i in $\{T_1, T_2, \dots, T_n\}$ solve the equation $r(t_\mathrm{f}) + \frac{v^2(t_\mathrm{f})}{2g} = \omega_\mathrm{f}$ and define the elements of the sample $X(m_0)$ of M_u as $X_i(m_0) := M_\mathrm{u}(T_i, m_0)$.

   (b) Build the KDE for the pdf of M_u using the SNR method (see 7.1.7) for computing the bandwidth.

$$\hat{f}_{m_0}(x) := \frac{1}{nh} \sum_{i=1}^{n} \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x - X_i(m_0)}{h}\right)^2}.$$

   (c) Compute $\hat{F}_{m_0}$ as

$$\hat{F}_{m_0}(\overline{m}_\mathrm{u}) := \int_0^{\overline{m}_\mathrm{u}} \hat{f}_{m_0}(x)dx.$$

For this example and all the others we will treat, the integral of the approximated density function is obtained numerically by using the composite Simpson's rule. Given an interval $[a, b]$, the integral of the function f is computed by dividing $[a, b]$ into an even number N of sub-intervals (in our case $N = 2000$) and applying the formula

$$\int_a^b f(x)dx \approx \frac{1}{3}\frac{b-a}{N}\left(f(a) + 2\sum_{i=1}^{\frac{N}{2}-1} f(x_{2i}) + 4\sum_{i=1}^{\frac{N}{2}} f(x_{2i-1}) + f(b)\right)$$

where

$$x_i := a + \frac{b-a}{N}i \quad \forall i \in \{0, 1, \ldots, N-1\}.$$

We decided to use Simpson's rule for this method because it is a good balance between ease of code implementation and precision, since the error of this quadrature formula is bounded by

$$\left(\frac{b-a}{N}\right)^4 (b-a) \max_{x\in[a,b]} |f^{(4)}(x)|.$$

Details and results on this formula can be found in [71].

The performances of the solver can be improved by discretizing an interval where the decision variable belongs (e.g. $[0, \approx 15.3]$, in this case) and store the values of the constraint function at the corresponding nodes prior to solving the problem. This way, each time the solver needs to evaluate the constraint function, it can interpolate it from the previously saved values instead of having to compute them.

3. **Solve the problem**
   Now that the solver knows how to compute the approximation $\hat{F}_{m_0}$ of F_{m_0}, we can solve problem (7.2.5) as a standard deterministic optimization problem by using WORHP as described in Subsection 7.2.1 with an initial guess equal to the solution of the deterministic problem.

Results

Constraint function Before solving problem (7.2.5), we want to check the regularity of $\hat{F}_{m_0}(\overline{m}_\mathrm{u})$ as a function of m_0. Figure 7.2.4 shows $\hat{F}_{m_0}(\overline{m}_\mathrm{u})$ evaluated at several values of m_0 for a sample of size $n = 500$. We can see that the set $\{m_0 \mid \hat{F}_{m_0}(\overline{m}_\mathrm{u}) < 1 - p\}$ is connected, meaning that we can expect the optimal solution to coincide with the left endpoint of the corresponding interval. The plot above seems to show that the constraint function is discontinuous at ≈ 1.04, but if we plot it only inside a small neighborhood of 1.04 we can see that it is not discontinuous but just very steep.

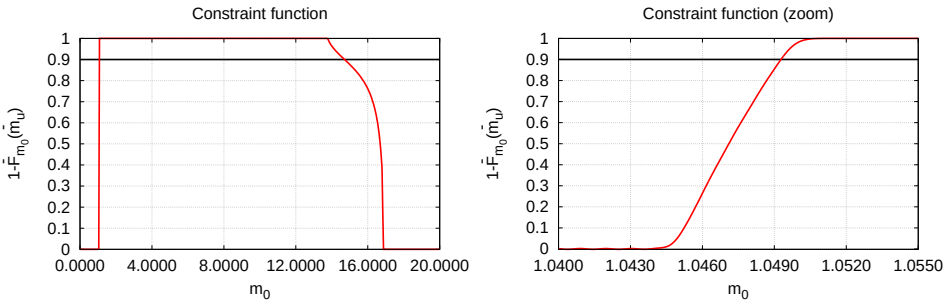


Figure 7.2.4: Approximation of $\mathbb{P}[M_{\mathrm{u}}(T, m_0) \geq \overline{m}_{\mathrm{u}}]$ as a function of m_0 for $n = 500$ (left). Zoom on the apparent discontinuity (right).

Convergence of approximated solutions Now that we have some information on where to find the optimal solution for a given n, we will solve problem (7.2.5) with WORHP for $n \in \{10, 20, \ldots, 500\}$.

Figure 7.2.5 shows the behavior of the sequence of optimal solutions m_0^* for all the considered values of n and the corresponding rate of success $R := \frac{N_{\mathrm{s}}}{N_{\mathrm{a}}}$ computed a posteriori with $N_{\mathrm{a}} = 10^5$. In figure 7.2.6 it can be seen

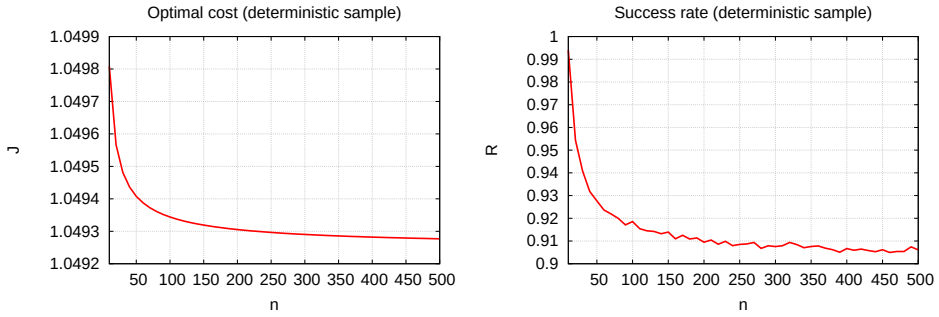


Figure 7.2.5: Plot of m_0^* and R as functions of n with a deterministic uniform sample from T.

instead how the results change if T is sampled randomly from its distribution instead of being sampled with uniform equidistant steps in its interval. Due to the random nature of the results of this kind of test, we show the results of ten simulations. Figure 7.2.7 contains the plots of the average values and variance of the ten simulations showed in figure 7.2.6. As expected, a smaller size of the sample X corresponds to a worse approximation of the chance constraint we want to satisfy. In this particular case the probability of success is over-estimated, leading to an increase of the optimal value for the initial mass m_0 due to the fact that the constraint the solver tries to satisfy is actually tighter.

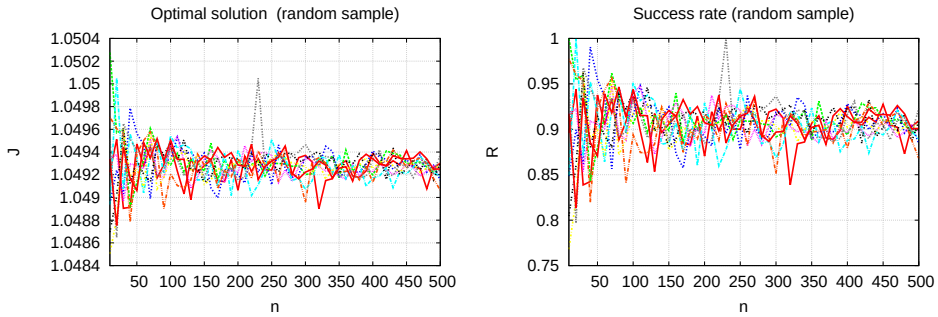


Figure 7.2.6: Plot of m_0^* and R as functions of n with a random sample from T (ten simulations).

For example, for $n = 500$ the optimal solution

$$m_0^* \approx 1.04928 \text{ [kg]}.$$

allows us to deliver the payload with a success rate $R \approx 90,73\%$ even if the maximum thrust T of the engine is subject to random uniform oscillations. Figure 7.2.8 show the corresponding plots.

Comparison with best/worst case scenario Table 7.2.5 compares the solution we just found for the stochastic optimization problem to the two solutions we obtain from the deterministic one if T is fixed at the values $(1 + \Delta\overline{T})\overline{T}$ and $(1 - \Delta\overline{T})\overline{T}$.

Case	T	m_0^* [kg]
Random	$\sim U(T_-, T_+)$	**1.04928**
Best	T_+	**1.04488**
Worst	T_-	**1.04984**

Table 7.2.5: Result comparison for extremal values of T.

We observe that the optimal mass of the stochastic problem is smaller than the one obtained in the worst deterministic case but bigger than the one of the best case.

Consistency An interesting comparison is the one between the solution of the deterministic problem (7.2.2) and its stochastic counterpart (7.2.3) when p and ΔT are respectively closer 1 and 0. We expect the two solutions to be similar, since we reduce the uncertainty on the random parameter and request the constraint on the payload to be satisfied with a higher probability.

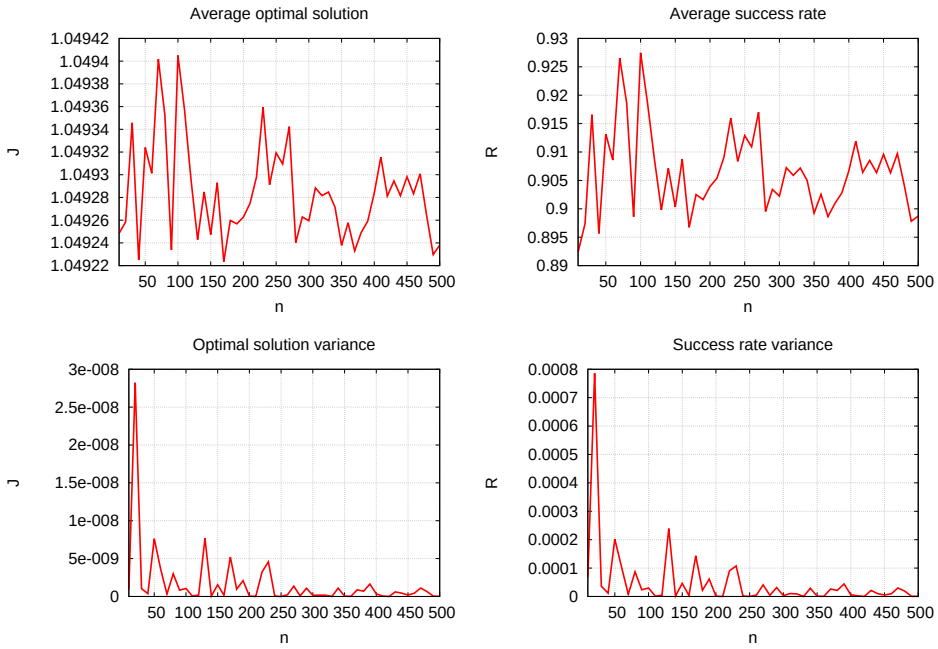


Figure 7.2.7: Plot of the average value and variance of m_0^* and R as functions of n with a random sample from T.

Table 7.2.2 shows that, for a high value of p, the optimal cost decreases with ΔT, suggesting that the solution we obtain from the stochastic optimization is consistent with the deterministic one. Unfortunately though, this method doesn't allow arbitrarily small values of ΔT. As reported in the table, when we don't provide enough variation to the sample, the success rate doesn't match the chosen probability. This is likely due to two issues related to the sample variance (see (7.1.7)), and therefore to ΔT. First, if ΔT is too small, the Gaussian distributions summed in (7.1.5) tend to superimpose over the same points and not spread on the real axis. This adds probability mass outside the domain of the distribution to be estimated. A negligible manifestation of this symptom can be observed even with $\Delta T = 0.1$ (see Figure 7.2.8, notice the white space beneath the red graph on the left and right sides of the vertical sample lines). Second, because the bandwidth depends on the sample variance (which itself depends on ΔT), the accuracy of the estimator might decrease if h is too small, as h appears as a denominator in (7.1.5).

SAHA vs. KDE

From the results in this section we can conclude that in this case the KDE performs much better than the SAHA. In this first example, the KDE is

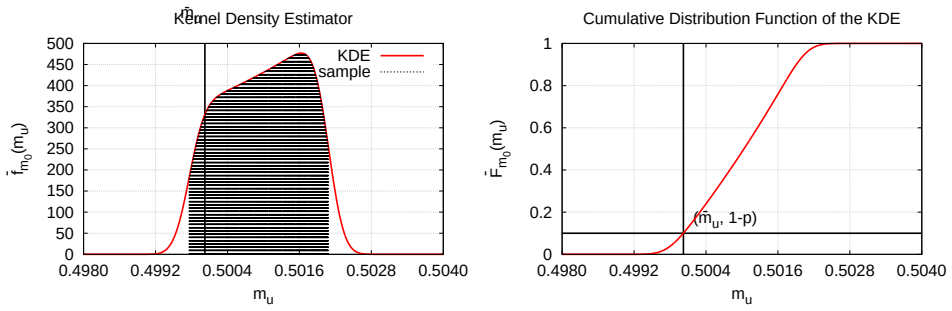


Figure 7.2.8: Plot of the Kernel Density Estimator $\hat{f}$ of $M_{\mathrm{u}}(T, m_0^*)$ and its integral $\hat{F}$.

able to provide an approximated solution to problem (7.2.3) with a rate of success which is very close to the desired one ($\approx 90.73\%$), while the best result provided by the SAHA ($\approx 95.06\%$), although feasible in terms of the satisfaction of the chance constraint, is still far from the 90% goal.

The accuracy of the probability constraint's satisfaction is not the only reason behind our preference of KDE over SAHA. The main issue with the application of the SAHA lies in the need of rewriting the constraint in expectation as a chance constraint, introducing irregularities in the model. Because this algorithm is designed to solve problems in the form (7.1.1), the required hypotheses supporting convergence might not be satisfied when defining the probability constraint by using the indicator function. Another drawback is the initialization: Table 7.2.4 shows how sensitive this algorithm is to the initial guess for approximated solution x_0, to the point that even initializing it with the solution of the deterministic counterpart (7.2.2) of problem (7.2.3) leads to a sequence converging to zero. In contrast to this, the same initialization for the KDE returns a very good approximation of the solution to the chance-constrained optimization problem. Even from the performance point of view the SAHA doesn't appear to be the best choice between the two: the necessary number of iterations needed by the algorithm to stabilize is very large (tens of thousands), compared to the iterations performed by WORHP with the KDE (a few dozen). Moreover, the choice of the parameters defining the SAHA is more difficult to perform compared to the KDE: in our case, for the first example we had to define both sequences $\{\varepsilon_k\}$ and $\{\rho_k\}$ as well as the sequence $\{r_k\}$ of smoothing parameters for the approximation of the indicator function; the parameterization of KDE, on the other hand, is mainly focused on one scalar quantity h (the bandwidth).

For all the aforementioned reasons, we will not use the SAHA in the next examples and we'll focus only on KDE.

n	p	ΔT	h	m_0^* [kg]	R
Stochastic					
500	0.8	0.5	0.00130	**1.05823**	**0.8047**
		0.25	0.00054	**1.05154**	**0.8036**
		0.1	0.00021	**1.04868**	**0.8014**
		0.05	0.00010	**1.04786**	**0.8015**
		0.025	0.00005	**1.04733**	**0.6874**
	0.9	0.5	0.00131	**1.06462**	**0.9031**
		0.25	0.00054	**1.05343**	**0.9037**
		0.1	0.00021	**1.04928**	**0.9050**
		0.05	0.00010	**1.04809**	**0.8881**
		0.025	0.00005	**1.04766**	**0.9520**
	0.995	0.5	0.00134	**1.07553**	**1.0000**
		0.25	0.00054	**1.05662**	**1.0000**
		0.1	0.00021	**1.05035**	**1.0000**
		0.05	0.00010	**1.04871**	**1.0000**
		0.025	0.00005	**1.04792**	**1.0000**
Deterministic					
—				1.04709	—

Table 7.2.6: Result comparison for different values of n, p and ΔT.

7.2.3 Test 2: Simple three stage launcher with three decision variables and three random variables

We now focus on a slightly more complex model, still in one dimension, involving a three-stage launcher. This type of space launchers is characterized by the fact that they are divided into three sections, each one with its own fuel load and engine. During the flight the launcher consumes progressively the fuel of each stage, and detaches the empty structure of one stage once its fuel load is depleted. In this model we have more agency on the dynamics, since we can choose the amount of initial fuel m_{ei} of each stage, meaning that now we have three decision variables. While the nature of the chance-constrained optimization problem remains the same, the number of random parameters has also been increased to three: we consider the thrust T_i of each stage engine to be random.

Model

The ODE system to be solved is the same as before and thus the solution can be obtained explicitly, but the introduction of discontinuities in the mass forces us to split the trajectory into three phases.

The initial mass of the launcher is defined as the sum of the three stages'

fuel and structure, plus the payload:

$$m(0) = \sum_{i=1}^{3}(1+k_i)m_{\mathrm{e}i} + \overline{m}_{\mathrm{u}}$$

where $\mathbf{k} = (k_1, k_2, k_3)$ and $\mathbf{m}_{\mathrm{e}} := (m_{\mathrm{e}1}, m_{\mathrm{e}2}, m_{\mathrm{e}3})$ are respectively the stage indexes and the fuel masses of the three stages. To simplify the notation we define the final time of each phase as

$$\begin{aligned} t_1 &:= \frac{v_{e1}m_{\mathrm{e}1}}{T_1} \\ t_2 &:= t_1 + \frac{v_{\mathrm{e}2}m_{\mathrm{e}2}}{T_2} \\ \bar{t}_3 &:= t_2 + \frac{v_{\mathrm{e}3}(m_{\mathrm{e}3} + \overline{m}_{\mathrm{u}})}{T_3}. \end{aligned}$$

We point out that, while t_1 and t_2 represent the exact duration of phases 1 and 2 (the launchers consumes all the fuel in the first and second stages), the quantity $\bar{t}_3$ is just an upper bound for the final time of the third phase. The final time of the third phase (and thus the trajectory) depends on both the total fuel mass and the energy constraint on the final orbit. In the definition of $\bar{t}_3$, the payload $\overline{m}_{\mathrm{u}}$ is summed to the fuel mass of the third stage, allowing the launcher to consume part of the payload in case the amount of fuel is not sufficient to satisfy the constraint on the final position. We will also define $\mathbf{T} := (T_1, T_2, T_3)$ and $\mathbf{v}_{\mathrm{e}} := (v_{\mathrm{e}1}, v_{\mathrm{e}2}, v_{\mathrm{e}3})$.

With these definitions, for $0 \leq t \leq t_1$ the solution to the ODE system is

$$\begin{aligned} r(t) &= \left(v_{\mathrm{e}1}t - \frac{v_{\mathrm{e}1}^2 m(0)}{T_1}\right)\ln\left(\frac{m(0)}{m(t)}\right) + v_{\mathrm{e}1}t - \frac{g}{2}t^2 \\ v(t) &= v_{\mathrm{e}1}\ln\left(\frac{m(0)}{m(t)}\right) - gt \\ m(t) &= \sum_{i=1}^{3}(1+k_i)m_{\mathrm{e}i} + \overline{m}_{\mathrm{u}} - \frac{T_1}{v_{\mathrm{e}1}}t \end{aligned}$$

otherwise, if $t_1 < t \leq t_2$

$$\begin{aligned} r(t) = &\left(v_{\mathrm{e}1}t - \frac{v_{\mathrm{e}1}^2 m(0)}{T_1}\right)\ln\left(\frac{m(0)}{m(t_1)}\right) + \\ &+ \left(v_{\mathrm{e}2}(t-t_1) - \frac{v_{\mathrm{e}2}^2(m(t_1) - k_1 m_{\mathrm{e}1})}{T_2}\right)\ln\left(\frac{m(t_1) - k_1 m_{\mathrm{e}1}}{m(t)}\right) + \\ &+ v_{\mathrm{e}1}t_1 + v_{\mathrm{e}2}(t-t_1) - \frac{g}{2}t^2 \\ v(t) = &\, v_{\mathrm{e}1}\ln\left(\frac{m(0)}{m(t_1)}\right) + v_{\mathrm{e}2}\ln\left(\frac{m(t_1) - k_1 m_{\mathrm{e}1}}{m(t)}\right) - gt \\ m(t) = &\sum_{i=2}^{3}(1+k_i)m_{\mathrm{e}i} + \overline{m}_{\mathrm{u}} - \frac{T_2}{v_{\mathrm{e}2}}(t-t_1) \end{aligned}$$

and lastly, if $t_2 < t \leq \bar{t}_3$

$$\begin{aligned}
r(t) = &\left(v_{\mathrm{e}1}t - \frac{v_{\mathrm{e}1}^2 m(0)}{T_1}\right)\ln\left(\frac{m(0)}{m(t_1)}\right) + \\
&+ \left(v_{\mathrm{e}2}(t-t_1) - \frac{v_{\mathrm{e}2}^2(m(t_1)-k_1 m_{\mathrm{e}1})}{T_2}\right)\ln\left(\frac{m(t_1)-k_1 m_{\mathrm{e}1}}{m(t_2)}\right) + \\
&+ \left(v_{\mathrm{e}3}(t-t_2) - \frac{v_{\mathrm{e}3}^2(m(t_2)-k_2 m_{\mathrm{e}2})}{T_3}\right)\ln\left(\frac{m(t_2)-k_2 m_{\mathrm{e}2}}{m(t)}\right) + \\
&+ v_{\mathrm{e}1}t_1 + v_{\mathrm{e}2}(t_2-t_1) + v_{\mathrm{e}3}(t-t_2) - \frac{g}{2}t^2 \\
v(t) = &\, v_{\mathrm{e}1}\ln\left(\frac{m(0)}{m(t_1)}\right) + v_{\mathrm{e}2}\ln\left(\frac{m(t_1)-k_1 m_{\mathrm{e}1}}{m(t_2)}\right) + \\
&+ v_{\mathrm{e}3}\ln\left(\frac{m(t_2)-k_2 m_{\mathrm{e}2}}{m(t)}\right) - gt \\
m(t) = &\, (1+k_3)m_{\mathrm{e}3} + \overline{m}_{\mathrm{u}} - \frac{T_3}{v_{\mathrm{e}3}}(t-t_2).
\end{aligned}$$

As we did in the previous section, let us first look at the solution of the deterministic optimization problem:

$$\begin{cases} \min\limits_{\mathbf{m}_{\mathrm{e}}\in\mathbb{R}^3_+} \sum\limits_{i=1}^{3}(1+k_i)m_{\mathrm{e}i} + \overline{m}_{\mathrm{u}} \\ M_{\mathrm{u}}(\mathbf{m}_{\mathrm{e}}) \geq \overline{m}_{\mathrm{u}} \end{cases} \tag{7.2.6}$$

where the constraint function is defined as

$$M_{\mathrm{u}}(\mathbf{m}_{\mathrm{e}}) := m(t_3(\mathbf{m}_{\mathrm{e}})) - k_3 m_{\mathrm{e}3}.$$

For a given $\mathbf{m}_{\mathrm{e}}$, the trajectory's final time $t_3(\mathbf{m}_{\mathrm{e}}) \in [t_2, \bar{t}_3]$ is the solution of the equation obtained by imposing the constraint $r(t_3) + \frac{v^2(t_3)}{2g} = \omega_{\mathrm{f}}$. Similarly to the previous example, this equation cannot be solved explicitly and its solution is computed numerically. Table 7.2.7 sums up the choice of parameters. The optimal solution found by WORHP is

Parameter	T_i	k_i	$v_{\mathrm{e}i}$	g	ω_{f}	$\overline{m}_{\mathrm{u}}$
Value	150 [N]	0.1	5 [m/s]	9.8 [m/s^2]	0.5 [m]	0.5 [kg]

Table 7.2.7: Parameters for the deterministic optimization

$$\begin{aligned}
m_{\mathrm{e}1}^* &\approx 0.21528 \text{ [kg]} \\
m_{\mathrm{e}2}^* &\approx 0.18378 \text{ [kg]} \\
m_{\mathrm{e}3}^* &\approx 0.07743 \text{ [kg]}.
\end{aligned}$$

with a corresponding optimal cost of

$$\sum_{i=1}^{3}(1+k_i)m_{\mathrm{e}i}^{*}+\overline{m}_{\mathrm{u}}\approx 1.02414\ [\mathrm{kg}].$$

Note that an optimal initial mass of ≈ 1.02414 [kg] is consistent with the one of the model with just one stage (≈ 1.04709 [kg]), since the whole point of separating launchers into stages is exactly to reduce its initial mass.

Figures 7.2.9 shows the corresponding optimal trajectory.

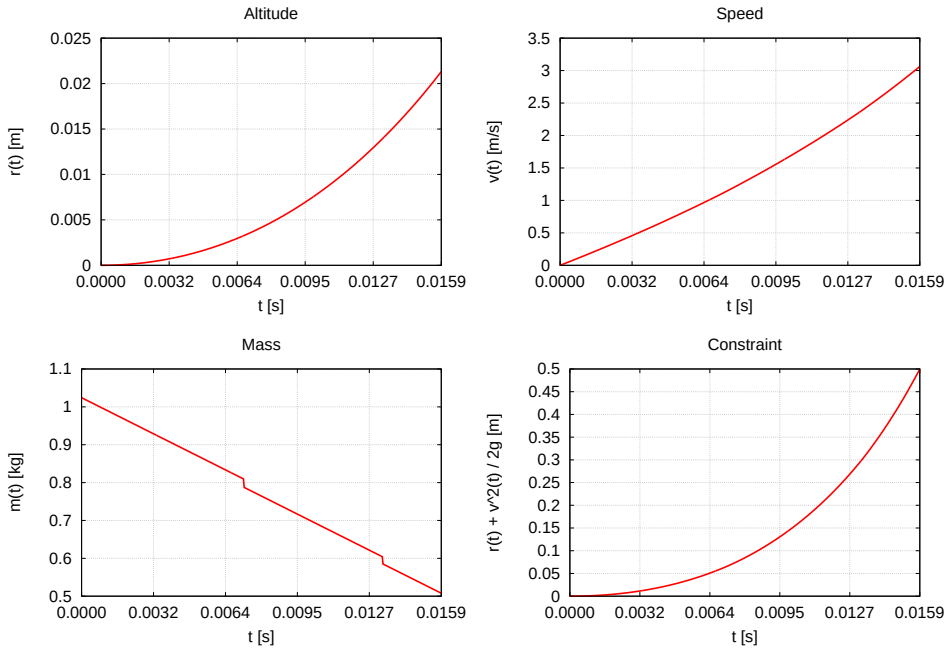


Figure 7.2.9: Plot of altitude, speed, mass and constraint for the three-stage launcher.

Problem statement

In this section we will consider the three-stage version of the stochastic optimization problem treated in the previous one.

$$\begin{cases}\min\limits_{\mathbf{m}_{\mathrm{e}}\in\mathbb{R}^3_+}\sum\limits_{i=1}^{3}(1+k_i)m_{\mathrm{e}i}+\overline{m}_{\mathrm{u}}\\ \mathbb{P}\big[M_{\mathrm{u}}(\mathbf{T},\mathbf{m}_{\mathrm{e}})\geq\overline{m}_{\mathrm{u}}\big]\geq p.\end{cases}\tag{7.2.7}$$

Each T_i is a uniformly distributed random variable on $[T_{i-},T_{i+}]$ with expected value $\overline{T}_i$:

$$T_i\sim U(T_{i-},T_{i+})$$

where $T_{i-} := \overline{T}_i(1 - \Delta T_i)$, $T_{i+} := \overline{T}_i(1 + \Delta T_i)$. We also define $\overline{\mathbf{T}} := (\overline{T}_1, \overline{T}_2, \overline{T}_3)$ and $\Delta\mathbf{T} := (\Delta T_1, \Delta T_2, \Delta T_3)$. The pdf of each element of the array $\mathbf{T}$ is

$$\phi_i(x) := \begin{cases} \frac{1}{T_{i+} - T_{i-}} & x \in [T_{i-}, T_{i+}] \\ 0 & \text{else} \end{cases} \quad \forall i \in \{1, 2, 3\}.$$

$M_{\mathrm{u}}(\mathbf{T}, \mathbf{m}_{\mathrm{e}})$ is a function of the random variables T_1, T_2 and T_3, parameterized by $\mathbf{m}_{\mathrm{e}}$:

$$M_{\mathrm{u}}(\mathbf{T}, \mathbf{m}_{\mathrm{e}}) := m(t_3(\mathbf{T}, \mathbf{m}_{\mathrm{e}})) - k_3 m_{\mathrm{e}3}.$$

Table 7.2.8 shows the choice of parameters defined in this subsection.

Parameter	p	$\overline{T}_i$	ΔT
Value	0.9	150 [N]	0.1

Table 7.2.8: Additional parameters for the stochastic optimization

Solution via Kernel Density Estimation

Like the previous example, we have to reformulate the chance constraint showing its dependency on the CDF F of random variable M_{u}.

$$\begin{cases} \min\limits_{\mathbf{m}_{\mathrm{e}} \in \mathbb{R}^3_+} \sum\limits_{i=1}^{3} (1 + k_i) m_{\mathrm{e}i} + \overline{m}_{\mathrm{u}} \\ F_{\mathbf{m}_{\mathrm{e}}}(\overline{m}_{\mathrm{u}}) \leq 1 - p. \end{cases} \tag{7.2.8}$$

Where

$$F_{\mathbf{m}_{\mathrm{e}}}(\overline{m}_{\mathrm{u}}) := \int_0^{\overline{m}_{\mathrm{u}}} f_{\mathbf{m}_{\mathrm{e}}}(x) dx$$

For each value of $\mathbf{m}_{\mathrm{e}}$ we are able to produce an approximation $\hat{F}_{\mathbf{m}_{\mathrm{e}}}$ of $F_{\mathbf{m}_{\mathrm{e}}}$ via KDE by drawing a sample from the random array $\mathbf{T}$. Our problem becomes

$$\begin{cases} \min\limits_{\mathbf{m}_{\mathrm{e}} \in \mathbb{R}^3_+} \sum\limits_{i=1}^{3} (1 + k_i) m_{\mathrm{e}i} + \overline{m}_{\mathrm{u}} \\ \hat{F}_{\mathbf{m}_{\mathrm{e}}}(\overline{m}_{\mathrm{u}}) \leq 1 - p. \end{cases} \tag{7.2.9}$$

The procedure used for solving problem (7.2.9) is described in the following steps.

1. **Draw the sample**
   We take n_{s} equidistant values for each T_i in the interval $[T_{i-}, T_{i+}]$ such that
   $$T_{ij} = T_{i-} + (j - 1)\frac{T_{i+} - T_{i-}}{n - 1} \quad \forall j \in 1, \ldots, n_{\mathrm{s}}$$

We can represent this sample as a $3 \times n_\mathrm{s}$ matrix:

$$\begin{pmatrix} T_{11} & T_{12} & \dots & T_{1n_\mathrm{s}} \\ T_{21} & T_{22} & \dots & T_{2n_\mathrm{s}} \\ T_{31} & T_{32} & \dots & T_{3n_\mathrm{s}}. \end{pmatrix}$$

2. **Define the constraint function**
   Instruct the solver on how to associate any choice of fuel mass $\mathbf{m}_\mathrm{e}$ to the constraint function $\hat{F}_{\mathbf{m}_\mathrm{e}}$.

   (a) The number of all the possible combinations of T_1, T_2 and T_3 in the sample matrix is n_s^3, this means that the size of the sample for M_u will be $n := n_\mathrm{s}^3$. For each combination $\mathbf{T}_i$ in $\{\mathbf{T}_1, \mathbf{T}_2, \dots, \mathbf{T}_n\}$ solve the equation $r(t_3) + \frac{v^2(t_3)}{2g} = \omega_\mathrm{f}$ and define the elements of the sample $X(\mathbf{m}_\mathrm{e})$ of M_u as $X_i(\mathbf{m}_\mathrm{e}) := M_\mathrm{u}(\mathbf{T}_i, \mathbf{m}_\mathrm{e})$.

   (b) Build the KDE for the pdf of M_u using the SNR method (see 7.1.7) for computing the bandwidth.

$$\hat{f}_{\mathbf{m}_\mathrm{e}}(x) := \frac{1}{nh} \sum_{i=1}^{n} \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x - X_i(\mathbf{m}_\mathrm{e})}{h}\right)^2}.$$

   (c) Compute $\hat{F}_{\mathbf{m}_\mathrm{e}}$ as

$$\hat{F}_{\mathbf{m}_\mathrm{e}}(\overline{m}_\mathrm{u}) := \int_0^{\overline{m}_\mathrm{u}} \hat{f}_{\mathbf{m}_\mathrm{e}}(x) dx.$$

3. **Solve the problem**
   Now that the solver knows how to compute the approximation $\hat{F}_{\mathbf{m}_\mathrm{e}}$ of $F_{\mathbf{m}_\mathrm{e}}$, we can solve problem (7.2.9) as a standard deterministic optimization problem by using WORHP as described in Subsection 7.2.1 with an initial guess equal to the solution of the deterministic problem.

Results

Convergence of approximated solutions Figures 7.2.10 shows the sequence of optimal costs for $n_\mathrm{s} \in \{2, 3, \dots, 20\}$ and the corresponding rates obtained by selecting a uniform sample from the variables T_i. Figure 7.2.11 instead, shows ten sequences of optimal costs and success rates obtained by drawing a random sample from the variables T_i. The average value and variance of the ten sequences can be seen in figure 7.2.12. For example, for $n_\mathrm{s} = 20$ (i.e. $n = n_\mathrm{s}^3 = 8000$) the optimal solution

$$\begin{aligned} m_{\mathrm{e}1}^* &\approx 0.20927 \text{ [kg]} \\ m_{\mathrm{e}2}^* &\approx 0.17057 \text{ [kg]} \\ m_{\mathrm{e}3}^* &\approx 0.09747 \text{ [kg]}. \end{aligned}$$

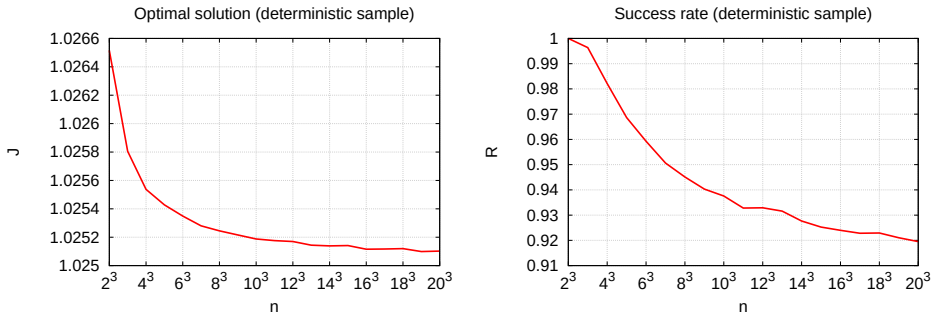


Figure 7.2.10: Plot of the optimal cost J and R as functions of n with a deterministic uniform sample from T.

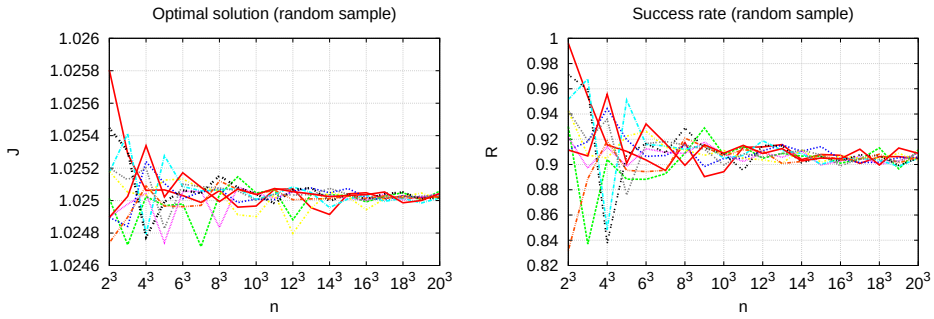


Figure 7.2.11: Plot of the optimal cost J and R as functions of n with a random sample from T (ten simulations).

with a corresponding optimal cost of

$$\sum_{i=1}^{3}(1+k_i)m_{\mathrm{e}i}^{*}+\overline{m}_{\mathrm{u}}\approx 1.02504\ [\mathrm{kg}].$$

allows us to deliver the payload with a success rate $R\approx 92\%$ even if the maximum thrust T_i of each stage engine is subject to random uniform oscillations. Figure 7.2.13 shows the corresponding plots.

Comparison with best/worst case scenario Table 7.2.9 compares the solution we just found for the stochastic optimization problem to the two solution we obtain from the deterministic one in the best and worst case.

We observe again that the optimal mass of the stochastic problem is smaller than the one obtained in the worst deterministic case but bigger than the one of the best case.

Consistency For this second example we repeat the comparison between the solution of the deterministic problem (7.2.6) and its stochastic counter-

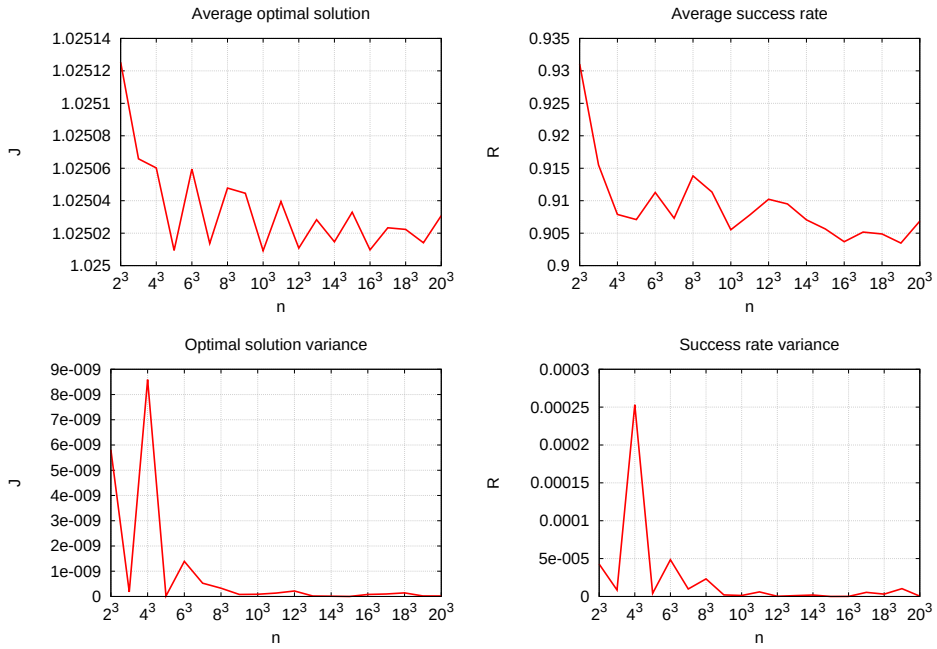


Figure 7.2.12: Plot of the average value and variance of optimal cost J and R as functions of n.

Case	T_i	m_0^* [kg]
Random	$\sim U(T_{i-}, T_{i+})$	**1.02504**
Best	T_{i+}	**1.02197**
Worst	T_{i-}	**1.02631**

Table 7.2.9: Result comparison for estremal values of $\mathbf{T}$.

part (7.2.7) when p and ΔT are respectively closer 1 and 0. The results in Table 7.2.10 are similar to the ones from the previous example.

Comparison between WORHP and IPOPT

We attempted to solve both deterministic and stochastic problems described in examples 1 and 2 by using the two solvers WORHP and IPOPT (see Subsection 7.2.1 for more details). In our case WORHP behaved much better than IPOPT (see table 7.2.11) and thus we decided to use the former to solve all the following examples. In the stochastic case, the behavior of both solvers depends heavily on the initial guess for the solution: some initializations can negatively affect performances and even sometimes prevent convergence. We are able to overcome this issue by initializing the solvers with the solution of the deterministic problem, which can be easily obtained.

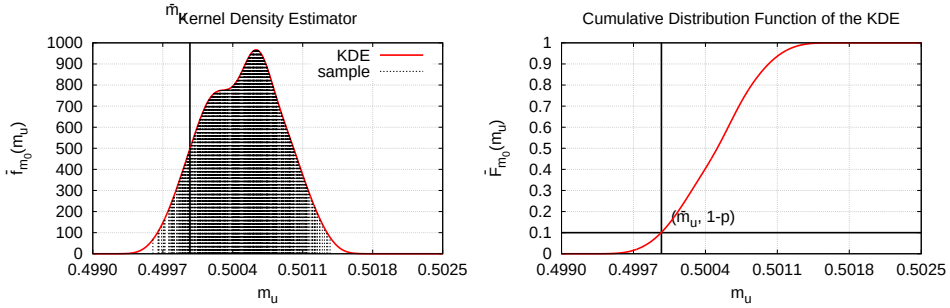


Figure 7.2.13: Plot of the Kernel Density Estimator $\hat{f}$ of $M_{\mathrm{u}}(\mathbf{T}, \mathbf{m}_{\mathrm{e}}^*)$ and its integral $\hat{F}$.

n	p	ΔT_i	h	$m_{\mathrm{e}1}$	$m_{\mathrm{e}2}$	$m_{\mathrm{e}3}$	m_0^* [kg]	R
Stochastic								
20^3	0.8	0.5	0.00043	0.216	0.171	0.096	**1.03090**	**0.7968**
		0.25	0.00018	0.206	0.175	0.097	**1.02642**	**0.7993**
		0.1	0.00007	0.209	0.170	0.098	**1.02464**	**0.8063**
		0.05	0.00003	0.207	0.158	0.112	**1.02394**	**0.7691**
		0.025	0.00002	0.216	0.168	0.092	**1.02406**	**0.8726**
	0.9	0.5	0.00043	0.216	0.174	0.096	**1.03454**	**0.9014**
		0.25	0.00018	0.202	0.180	0.097	**1.02748**	**0.9020**
		0.1	0.00007	0.208	0.171	0.098	**1.02501**	**0.9032**
		0.05	0.00003	0.211	0.166	0.099	**1.02426**	**0.8770**
		0.025	0.00002	0.209	0.179	0.103	**1.04045**	**1.0000**
	0.995	0.5	0.00043	0.217	0.180	0.096	**1.04237**	**0.9955**
		0.25	0.00018	0.200	0.185	0.097	**1.02997**	**0.9957**
		0.1	0.00007	0.210	0.171	0.097	**1.02590**	**0.9989**
		0.05	0.00003	0.212	0.167	0.098	**1.02476**	**0.9996**
		0.025	0.00002	0.209	0.170	0.097	**1.02425**	**1.0000**
Deterministic								
—				0.215	0.184	0.077	1.02414	—

Table 7.2.10: Result comparison for different values of n, p and ΔT.

Solver	Example	Problem	Convergence	Iterations	Time [s]
WORHP →	1 →	Det. →	Yes	< 5	< 1
		Sto. →	Yes	$5 \div 10$	< 1
	2 →	Det. →	Yes	< 5	< 1
		Sto. →	Yes	< 5	$1 \div 5$
IPOPT →	1 →	Det. →	Yes	$25 \div 50$	$5 \div 10$
		Sto. →	Yes	$50 \div 100$	$10 \div 15$
	2 →	Det. →	No	—	—
		Sto. →	No	—	—

Table 7.2.11: Comparison between WORHP and IPOPT

7.2.4 Test 3: Simple three stage launcher with three decision variables and nine random variables

The next step in complexity consists in introducing new uncertainties in the model. While in the previous example we fixed all the parameters but the thrusts $\mathbf{T}$, in this one we will also consider the stage indexes $\mathbf{k}$ and the fuel velocities $\mathbf{v}_{\mathrm{e}}$ as random variables. We can skip the description of the dynamics since it coincides with the one of the previous section.

Problem statement

If we want to adapt the formulation of problem 7.2.7 to this model, we have to keep in mind that now the cost to be minimized also depends on the random array $\mathbf{k}$ and it has to be defined as an expectation.

$$\mathbb{E}\left[\sum_{i=1}^{3}(1+k_i)m_{\mathrm{e}i}+\overline{m}_{\mathrm{u}}\right]=\sum_{i=1}^{3}\left(1+\mathbb{E}\left[k_i\right]\right)m_{\mathrm{e}i}+\overline{m}_{\mathrm{u}}.$$

Now, since each k_i is a uniformly distributed random variable on the interval $[k_{i-}, k_{i+}]$ with expected value $\overline{k}_i$, we can write the cost as

$$\sum_{i=1}^{3}\left(1+\mathbb{E}\left[k_i\right]\right)m_{\mathrm{e}i}+\overline{m}_{\mathrm{u}}=\sum_{i=1}^{3}(1+\overline{k}_i)m_{\mathrm{e}i}+\overline{m}_{\mathrm{u}}.$$

This leads us to the stochastic optimization problem

$$\begin{cases}\min\limits_{\mathbf{m}_{\mathrm{e}}\in\mathbb{R}^3_+}\sum\limits_{i=1}^{3}(1+\overline{k}_i)m_{\mathrm{e}i}+\overline{m}_{\mathrm{u}}\\ \mathbb{P}\left[M_{\mathrm{u}}(\mathbf{T},\mathbf{k},\mathbf{v}_{\mathrm{e}},\mathbf{m}_{\mathrm{e}})\geq\overline{m}_{\mathrm{u}}\right]\geq p.\end{cases}\tag{7.2.10}$$

In this case we have a total of nine uniform random variables (three random arrays of dimension three): $\mathbf{T}$, $\mathbf{k}$ and $\mathbf{v}_{\mathrm{e}}$. We define the arrays $\overline{\mathbf{k}}$, $\overline{\mathbf{v}}_{\mathrm{e}}$, $\Delta\mathbf{k}$ and $\Delta\mathbf{v}_{\mathrm{e}}$ in the same way we previously did for $\mathbf{T}$.

$M_{\mathrm{u}}(\mathbf{T},\mathbf{k},\mathbf{v}_{\mathrm{e}},\mathbf{m}_{\mathrm{e}})$ is a function of the random arrays $\mathbf{T}$, $\mathbf{k}$ and $\mathbf{v}_{\mathrm{e}}$, parameterized by $\mathbf{m}_{\mathrm{e}}$:

$$M_{\mathrm{u}}(\mathbf{T},\mathbf{k},\mathbf{v}_{\mathrm{e}},\mathbf{m}_{\mathrm{e}}):=m\big(t_3(\mathbf{T},\mathbf{k},\mathbf{v}_{\mathrm{e}},\mathbf{m}_{\mathrm{e}})\big)-k_3m_{\mathrm{e}3}.$$

Table 7.2.12 shows the choice of parameters defined in this subsection.

Parameter	p	$\overline{T}_i$	ΔT_i	$\overline{k}_i$	Δk_i	$\overline{v}_{\mathrm{e}i}$	$\Delta v_{\mathrm{e}i}$
Value	0.9	150 [N]	0.1	0.1	0.1	5 [m/s]	0.1

Table 7.2.12: Additional parameters for the stochastic optimization

Solution via Kernel Density Estimation

As for the previous example, we have to reformulate the chance constraint showing its dependency on the CDF F of random variable M_{u}.

$$\begin{cases} \min\limits_{\mathbf{m}_{\mathrm{e}} \in \mathbb{R}^3_+} \sum\limits_{i=1}^{3} (1 + \overline{k}_i) m_{\mathrm{e}i} + \overline{m}_{\mathrm{u}} \\ F_{\mathbf{m}_{\mathrm{e}}}(\overline{m}_{\mathrm{u}}) \leq 1 - p. \end{cases} \tag{7.2.11}$$

Where

$$F_{\mathbf{m}_{\mathrm{e}}}(\overline{m}_{\mathrm{u}}) := \int_0^{\overline{m}_{\mathrm{u}}} f_{\mathbf{m}_{\mathrm{e}}}(x) dx$$

For each value of $\mathbf{m}_{\mathrm{e}}$ we are able to produce an approximation $\hat{F}_{\mathbf{m}_{\mathrm{e}}}$ of $F_{\mathbf{m}_{\mathrm{e}}}$ via KDE by drawing a sample from the random arrays $\mathbf{T}$, $\mathbf{k}$ and $\mathbf{v}_{\mathrm{e}}$. Our problem becomes

$$\begin{cases} \min\limits_{\mathbf{m}_{\mathrm{e}} \in \mathbb{R}^3_+} \sum\limits_{i=1}^{3} (1 + \overline{k}_i) m_{\mathrm{e}i} + \overline{m}_{\mathrm{u}} \\ \hat{F}_{\mathbf{m}_{\mathrm{e}}}(\overline{m}_{\mathrm{u}}) \leq 1 - p \end{cases} \tag{7.2.12}$$

The procedure used for solving problem (7.2.12) is described in the following steps.

1. **Draw the sample**
   Take a sample of size n from the random array $(\mathbf{T}, \mathbf{k}, \mathbf{v}_{\mathrm{e}})$:

   $$\{(\mathbf{T}_1, \mathbf{k}_1, \mathbf{v}_{\mathrm{e}1}), (\mathbf{T}_2, \mathbf{k}_2, \mathbf{v}_{\mathrm{e}2}), \ldots, (\mathbf{T}_n, \mathbf{k}_n, \mathbf{v}_{\mathrm{e}n})\}.$$

2. **Define the constraint function**
   Instruct the solver on how to associate any choice of fuel mass $\mathbf{m}_{\mathrm{e}}$ to the constraint function $\hat{F}_{\mathbf{m}_{\mathrm{e}}}$.

   (a) For each element $(\mathbf{T}_i, \mathbf{k}_i, \mathbf{v}_{\mathrm{e}i})$ solve the equation $r(t_3) + \frac{v^2(t_3)}{2g} = \omega_{\mathrm{f}}$ and define $X_i(\mathbf{m}_{\mathrm{e}}) := M_{\mathrm{u}}(\mathbf{T}_i, \mathbf{k}_i, \mathbf{v}_{\mathrm{e}i}, \mathbf{m}_{\mathrm{e}})$.

   (b) Build the KDE for the pdf of M_{u} using the SNR method (see 7.1.7) for computing the bandwidth.

   $$\hat{f}_{\mathbf{m}_{\mathrm{e}}}(x) := \frac{1}{nh} \sum_{i=1}^{n} \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x - X_i(\mathbf{m}_{\mathrm{e}})}{h}\right)^2}.$$

   (c) Compute $\hat{F}_{\mathbf{m}_{\mathrm{e}}}$ as

   $$\hat{F}_{\mathbf{m}_{\mathrm{e}}}(\overline{m}_{\mathrm{u}}) := \int_0^{\overline{m}_{\mathrm{u}}} \hat{f}_{\mathbf{m}_{\mathrm{e}}}(x) dx.$$

3. **Solve the problem**
   Now that the solver knows how to compute the approximation $\hat{F}_{\mathbf{m}_\mathrm{e}}$ of $F_{\mathbf{m}_\mathrm{e}}$, we can solve problem (7.2.12) as a standard deterministic optimization problem by using WORHP as described in Subsection 7.2.1 with an initial guess equal to the solution of the deterministic problem.

Results

Convergence of approximated solutions The results presented in this example have been obtained by using exclusively random samples of the random parameters. Figures 7.2.14 show the behavior of ten sequences of optimal costs for $n \in \{100, 200, \ldots, 10000\}$ and the corresponding rates of success $R := \frac{N_\mathrm{s}}{N_\mathrm{a}}$ computed a posteriori with $N_\mathrm{a} = 10^5$. Figure 7.2.15 instead

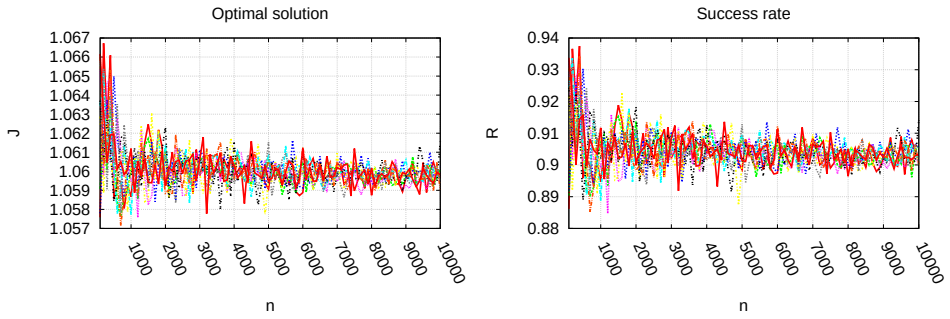


Figure 7.2.14: Plot of the optimal cost J and R as functions of n (ten simulations).

shows the the average value and variance of the ten sequences previously shown for each n.

For example for $n = 500$ the optimal solution

$$\begin{aligned} m_{\mathrm{e}1}^* &\approx 0.22222 \text{ [kg]} \\ m_{\mathrm{e}2}^* &\approx 0.18356 \text{ [kg]} \\ m_{\mathrm{e}3}^* &\approx 0.10289 \text{ [kg]}. \end{aligned}$$

with a corresponding optimal cost of

$$\sum_{i=1}^{3} (1 + \overline{k}_i) m_{\mathrm{e}i}^* + \overline{m}_\mathrm{u} \approx 1.05953 \text{ [kg]}.$$

allows us to deliver the payload $\overline{m}_\mathrm{u} = 0.5$ with a success rate $R \approx 90\%$ even if the maximum thrust T_i, the stage index k_i and the fuel speed $v_{\mathrm{e}i}$ of each stage are subject to random uniform oscillations. Figures 7.2.16 shows the related plots.

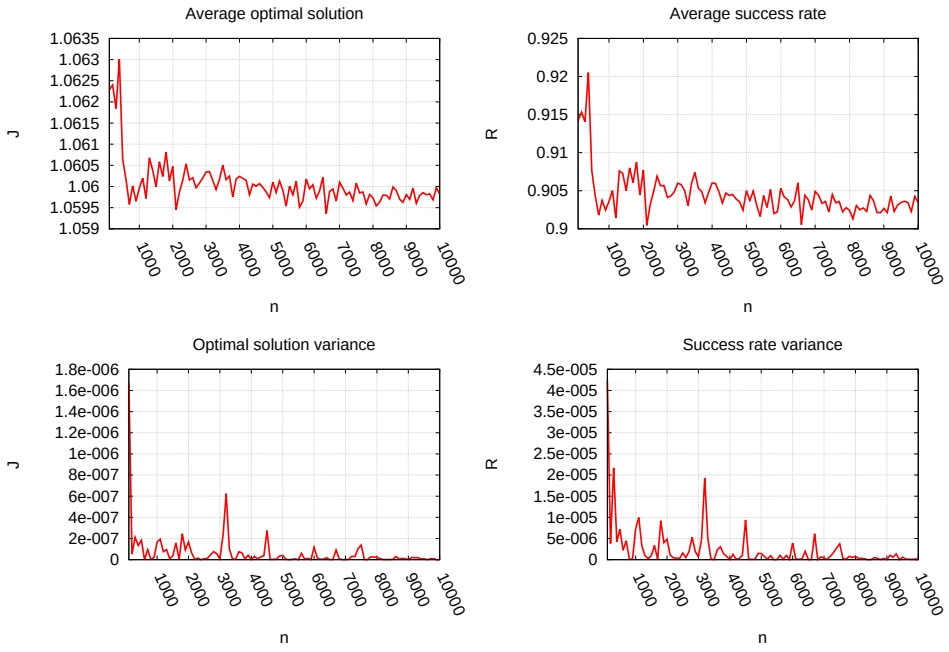


Figure 7.2.15: Plot of the average value and variance of optimal cost J and R as functions of n.

Comparison with best/worst case scenario Table 7.2.13 compares the solution we just found for the stochastic optimization problem to the two solution we obtain from the deterministic one in the best and worst case.

Case	T_i	k_i	v_{ei}	m_0^* [kg]
Random	$\sim U(T_{i-}, T_{i+})$	$\sim U(k_{i-}, k_{i+})$	$\sim U(v_{ei-}, v_{ei+})$	**1.05953**
Best	T_{i+}	k_{i-}	v_{ei+}	**0.94971**
Worst	T_{i-}	k_{i+}	v_{ei-}	**1.12464**

Table 7.2.13: Result comparison for extremal values of $\mathbf{T}$, $\mathbf{k}$ and $\mathbf{v}_e$.

We observe again that the optimal mass of the stochastic problem is smaller than the one obtained in the worst deterministic case but bigger than the one of the best case.

Consistency Table 7.2.14 shows the comparison between the solution of the deterministic problem (7.2.6) and its stochastic counterpart (7.2.10) when p is close to 1 and ΔT_i, Δk_i and Δv_{ei} are close to 0. In contrast with the previous example, the results showed in the table confirm that it is possible to reduce the variance of the random variables if their number is high enough to grant a sparse sample. In this example the bandwidth

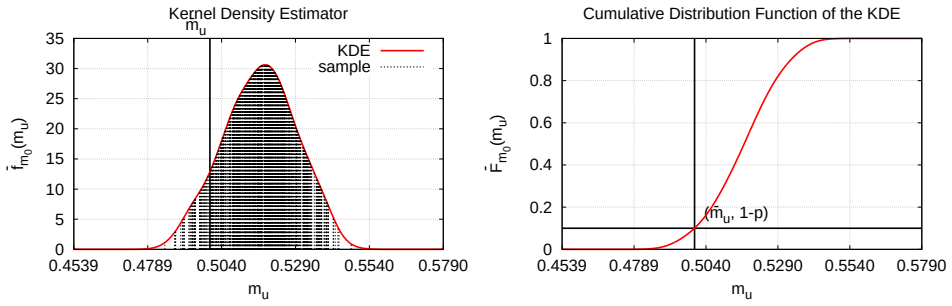


Figure 7.2.16: Plot of the Kernel Density Estimator $\hat{f}$ of $M_{\mathrm{u}}(T, \mathbf{m}_{\mathrm{e}}^*)$ and its integral $\hat{F}$.

reaches the value ≈ 0.00002 with a 0.1% variation of the random parameters while, for the previous example, the same bandwidth value was obtained with a 2.5% variation.

n	p	ΔT_i Δk_i Δv_{ei}	h	m_{e1}	m_{e2}	m_{e3}	m_0^* [kg]	R
Stochastic								
10^4	0.8	0.5	0.01169	0.218	0.191	0.202	**1.17137**	**0.7966**
		0.25	0.00508	0.226	0.185	0.124	**1.08867**	**0.8120**
		0.1	0.00191	0.237	0.158	0.103	**1.04773**	**0.8055**
		0.01	0.00019	0.214	0.166	0.098	**1.02593**	**0.8024**
		0.001	0.00002	0.209	0.167	0.101	**1.02389**	**0.8281**
	0.9	0.5	0.01472	0.186	0.281	0.271	**1.31146**	**0.9052**
		0.25	0.00517	0.227	0.205	0.133	**1.12153**	**0.8993**
		0.1	0.00192	0.218	0.184	0.106	**1.05956**	**0.9018**
		0.01	0.00018	0.215	0.166	0.098	**1.02707**	**0.9013**
		0.001	0.00002	0.214	0.166	0.097	**1.02410**	**0.9397**
	0.995	0.5	no convergence					
		0.25	0.00579	0.261	0.214	0.197	**1.23939**	**0.9960**
		0.1	0.00202	0.220	0.212	0.106	**1.09191**	**0.9951**
		0.01	0.00019	0.216	0.168	0.098	**1.02984**	**0.9959**
		0.001	0.00002	0.212	0.169	0.096	**1.02441**	**0.9996**
Deterministic								
—				0.215	0.184	0.077	1.02414	—

Table 7.2.14: Result comparison for different values of n, p and ΔT.

7.2.5 Test 4: Simple single stage launcher with continuous control and one random variable

This example takes inspiration from the first one by using the same single-stage model, the difference is that now we consider an optimal control problem.

The motion of the launcher retains the same nature of a one-dimensional vertical ascent, but this time we can act on the dynamics at each instant t. Contrary to all the previous examples in which the optimization variables directly influence only the initial state of the launcher, we can now control the vehicle at any point of the trajectory. We introduce the variable $u(t) \in [0, 1]$ which represents the percentage of maximum thrust we want to use at a given time t, meaning that for every t the thrust force and the fuel consumption are equal to $Tu(t)$ and $\frac{T}{v_e}u(t)$ respectively. We are again considering the case where only the parameter T is a random variable but now our objective is to maximize the final mass of the launcher while making sure that the launcher's final altitude is higher than a given value r_f with a probability of at least p.

A solution to this problem consists in an optimal control function $u^* : \mathbb{R}_+ \to [0, 1]$ such that, if we apply u^* regardless of the value of T, the probability of the final altitude being greater than r_f is at least p.

Model

The modified ODE system is

$$\begin{cases} \dot{r}(t) = v(t) & t \in (0, t_\mathrm{f}] \\ \dot{v}(t) = \frac{T}{m(t)} u(t) - g & t \in (0, t_\mathrm{f}] \\ \dot{m}(t) = -\frac{T}{v_\mathrm{e}} u(t) & t \in (0, t_\mathrm{f}] \\ r(0) = 0 \\ v(0) = 0 \\ m(0) = m_0 \end{cases}$$

where

$$m_0 := (1 + k) m_\mathrm{e} + m_\mathrm{u}$$

is the initial mass. The control function u belongs to $\mathcal{U}$, where

$$\mathcal{U} := \{u : \mathbb{R}_+ \to [0, 1] \subset \mathbb{R} \quad | \quad u \text{ is measurable}\}.$$

For the purpose of this test, we will integrate the equations numerically with the standard fourth-order Runge-Kutta method. The continuous control is thus replaced by a piece-wise constant function. If we denote with n_t the number of time steps, we can identify a control strategy u with the array of values taken at each time step:

$$u := (u_1, u_2, \ldots, u_{n_t}) \in \mathbb{R}_+^{n_t}.$$

Before defining our stochastic optimization problem, let us first look at the solution to its deterministic counterpart:

$$\begin{cases} \max\limits_{u \in \mathcal{U}} m(t_\mathrm{f}) \\ r(t_\mathrm{f}) \geq r_\mathrm{f} \end{cases} \tag{7.2.13}$$

which, implementing the numerical integration described above, is approximated by

$$\begin{cases} \max\limits_{u \in [0,1]^{n_t}} m(t_\mathrm{f}) \\ r(t_\mathrm{f}) \geq r_\mathrm{f}. \end{cases}$$

Table 7.2.15 sums up the choice of parameters for this model. The optimal cost found by WORHP is

$$m(t_\mathrm{f}) \approx 4.6141 \text{ [kg]}.$$

Figures 7.2.17 shows the corresponding optimal trajectory. We point out that, up to discretization errors, the computed deterministic control is bang-bang.

Parameter	T	k	v_e	g
Value	150 [N]	0.1	5 [m/s]	9.8 [m/s^2]

Parameter	m_e	m_u	r_f	t_f	n_t
Value	7.5 [kg]	0.5 [kg]	0.2 [m]	0.2 [s]	100

Table 7.2.15: Parameters for the deterministic optimization

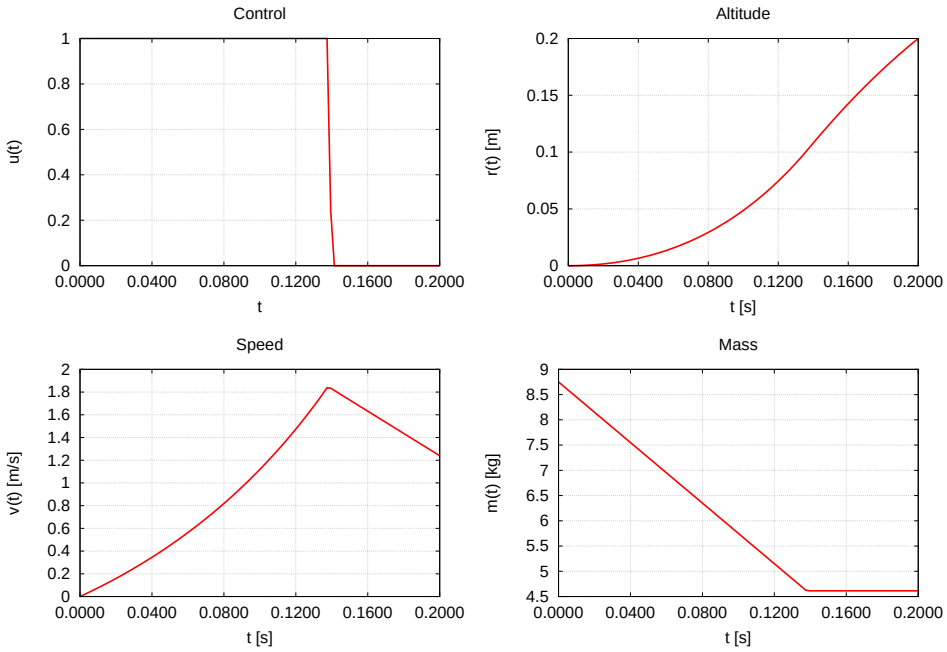


Figure 7.2.17: Plot of control, altitude, speed and mass for the controlled single-stage launcher.

Problem statement

Our goal is to reach at least the altitude r_f with a 90% probability while maximizing the final mass of the launcher. If we want to adapt the formulation of problem 3.1.1 to this model, we have to keep in mind that the cost to be minimized also depends on the random parameter T and it has to be defined as an expectation.

$$\mathbb{E}\left[m(t_f)\right] = \mathbb{E}\left[\int_0^{t_f} m_0 - \frac{T}{v_e}u(t)dt\right] = \int_0^{t_f} m_0 - \frac{\mathbb{E}[T]}{v_e}u(t)dt.$$

Now, since T is a uniformly distributed random variable on the interval $[T_-, T_+]$ with expected value $\overline{T}$, we can define the cost as

$$\overline{m}(t_f) := \int_0^{t_f} m_0 - \frac{\overline{T}}{v_e}u(t)dt.$$

This leads us to the stochastic optimization problem

$$\begin{cases} \max\limits_{u \in \mathcal{U}} \overline{m}(t_\mathrm{f}) \\ \mathbb{P}\big[R_\mathrm{f}(T,u) \geq r_\mathrm{f}\big] \geq p \end{cases} \tag{7.2.14}$$

where $R_\mathrm{f}(T,u)$ is the final altitude as a function of the random variable T, parameterized by u.

Table 7.2.16 shows the choice of parameters defined in this subsection.

Parameter	p	$\overline{T}$	ΔT
Value	0.9	150 [N]	0.1

Table 7.2.16: Additional parameters for the stochastic optimization

Solution via Kernel Density Estimation

In order to use the KDE we have to reformulate the chance constraint showing its dependency on the CDF F of random variable $r_u(T,t_\mathrm{f})$. Let $f_{t_\mathrm{f},u}$ be the pdf of r_u, parameterized by t_f. From the definition of $f_{t_\mathrm{f},u}$ we can rewrite problem 7.2.14 as

$$\begin{cases} \max\limits_{u \in \mathcal{U}} \overline{m}(t_\mathrm{f}) \\ F_u(r_\mathrm{f}) \leq 1-p. \end{cases} \tag{7.2.15}$$

For each choice of u we are able to produce an approximation $\hat{F}_u$ of F_u via KDE by drawing a sample of size from the random variable T. Our problem becomes

$$\begin{cases} \max\limits_{u \in \mathcal{U}} \overline{m}(t_\mathrm{f}) \\ \hat{F}_u(r_\mathrm{f}) \leq 1-p. \end{cases} \tag{7.2.16}$$

The procedure used for solving problem (7.2.16) is described in the following steps.

1. **Draw the sample**
   Take n equidistant values for T in the interval $[T_-, T_+]$ such that

   $$T_i = T_- + (i-1)\frac{T_+ - T_-}{n-1} \quad \forall i \in 1,\dots,n.$$

2. **Define the constraint function**
   Instruct the solver on how to associate a control u to the constraint function $\hat{F}_u$.

   (a) For each T_i in $\{T_1, T_2, \dots, T_n\}$ define the elements of the sample $X(u)$ of R_f as $X_i(u) := R_\mathrm{f}(T_i, u)$.

(b) Build the KDE for the pdf of R_f using the SNR method (see 7.1.7) for computing the bandwidth.

$$\hat{f}_u(x) := \frac{1}{nh}\sum_{i=1}^{n}\frac{1}{\sqrt{2\pi}}e^{-\frac{1}{2}\left(\frac{x-X_i(u)}{h}\right)^2}.$$

(c) Compute $\hat{F}_u$ as

$$\hat{F}_u(r_\mathrm{f}) := \int_0^{r_\mathrm{f}} \hat{f}_u(x)dx.$$

3. **Solve the problem**
Now that the solver knows how to compute the approximation $\hat{F}_u$ of F_u, we can solve problem (7.2.16) as a standard deterministic optimization problem by using WORHP as described in Subsection 7.2.1 with an initial guess equal to the solution of the deterministic problem.

Results

Convergence of approximated solutions Figures 7.2.18 shows the behavior of the sequence of optimal costs for $n \in \{10, 20, 30, \ldots, 500\}$ and the corresponding rate of success $R := \frac{N_\mathrm{s}}{N_\mathrm{a}}$ computed a posteriori with $N_\mathrm{a} = 10^5$. For example, for $n = 500$ the optimal cost and the success rate are

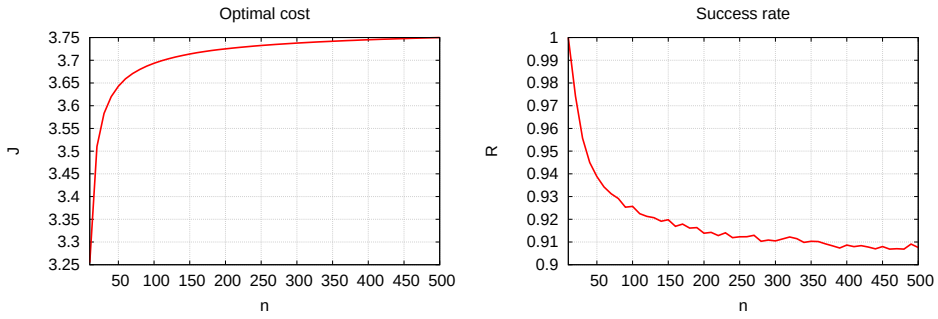


Figure 7.2.18: Plot of $m(t_\mathrm{f})$ corresponding to u^* and R as functions of n.

$$\overline{m}(t_\mathrm{f}) \approx 3.1934 \text{ [kg]}$$

and $R = 91.06\%$. The corresponding optimal control u^* is shown in figure 7.2.19. Figure 7.2.20 shows the other related plots.

Comparison with best/worst case scenario Table 7.2.17 and figure 7.2.21 compare the solution we just found for the stochastic optimization problem to the two solution we obtain from the deterministic one in the best and worst cases. In this case the results are different from the previous cases: the optimal final mass of the stochastic problem is smaller than both the ones obtained in the best and worst deterministic cases.

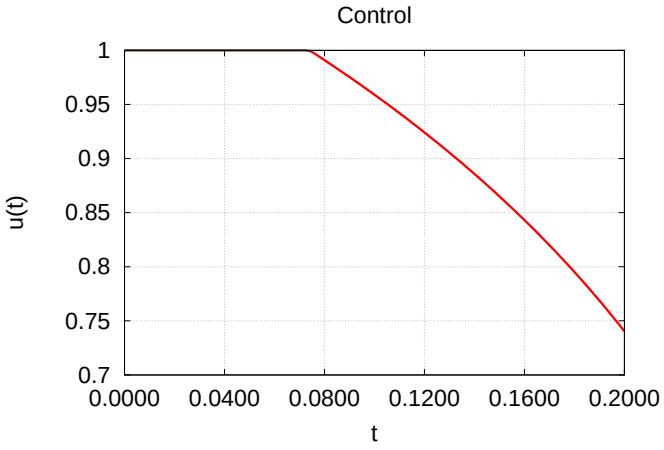


Figure 7.2.19: Optimal control for $n = 500$.

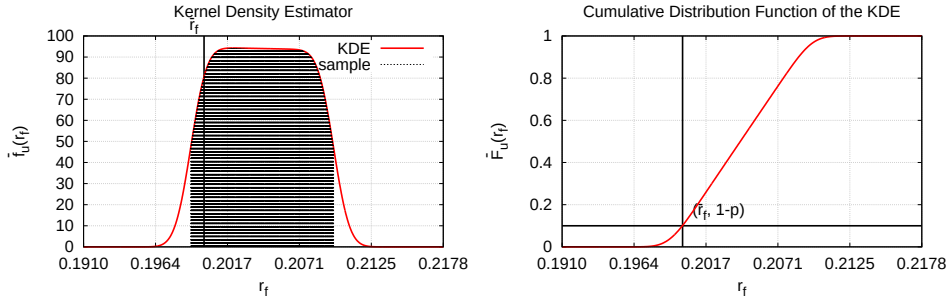


Figure 7.2.20: Plot of the Kernel Density Estimator $\hat{f}$ of $R_{\mathrm{f}}(T, u^*)$ and its integral $\hat{F}$.

Consistency Table 7.2.18 shows the comparison between the solution of the deterministic problem (7.2.13) and its stochastic counterpart (7.2.14) when p is close to 1 and T_i is close to 0. For the results in the table we set

$$u(t) = 1 \quad \forall t \in [0, t_{\mathrm{f}}]$$

as the initial guess for the optimal control strategy. We could also initialize u with the solution of the deterministic problem (7.2.16) but, surprisingly, the constant initialization is the one that allows WORHP to converge more often. Unfortunately the solver unable to converge to an optimal solution for every combination of p and ΔT. In the two cases $(p, \Delta T) = (0.8, 0.025)$ and $(p, \Delta T) = (0.995, 0.025)$ we tried, without success, three different initializations for u: the constant initialization previously described; the optimal solution of the deterministic problem and the initialization by continuation, i.e. initializing u with the solution of the problems with $(p, \Delta T) = (0.8, 0.05)$

Case	T	$\overline{m}(t_\mathrm{f})$ [kg]
Random	$\sim U(T_-, T_+)$	**3.19589**
Best	T_+	**4.88176**
Worst	T_-	**3.93085**

Table 7.2.17: Result comparison for extremal values of T.

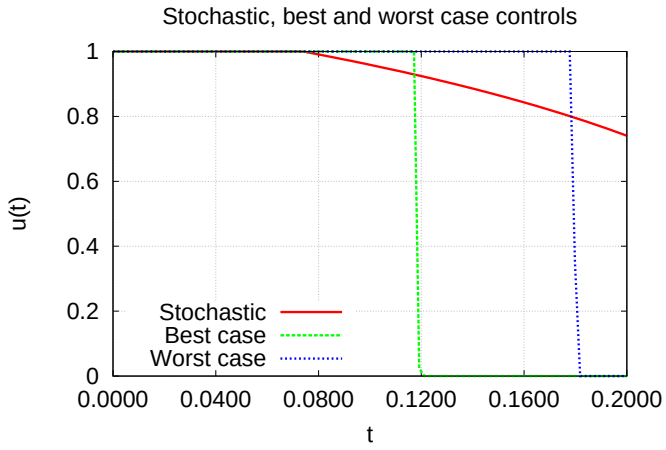


Figure 7.2.21: Comparison between stochastic, best and worst case controls.

and $(p, \Delta T) = (0.995, 0.05)$. We are aware that these three different initialization techniques are not exhaustive for the purpose of solving the convergence issues. For instance, the continuation method could be refined by introducing more intermediary steps. In any case, the best way to avoid such problems with solvers would be to implement from scratch a code for the solution of optimal control problems. Contrary to a non open source library like WORHP, this would allow the user to investigate in detail convergence issues and modify the algorithm accordingly by adapting the source code.

n	p	ΔT	h	$\overline{m}(t_\mathrm{f})$ [kg]	R
Stochastic					
500	0.8	0.5	no convergence		
		0.25	no convergence		
		0.1	0.01051	**4.04155**	**0.8012**
		0.05	0.00167	**7.17424**	**0.0000**
		0.025	no convergence		
	0.9	0.5	no convergence		
		0.25	no convergence		
		0.1	0.01118	**3.19345**	**0.9098**
		0.05	0.00506	**3.63893**	**0.9085**
		0.25	0.00122	**7.11364**	**0.0000**
	0.995	0.5	no convergence		
		0.25	no convergence		
		0.1	0.01223	**2.75000**	**1.0000**
		0.05	0.00536	**3.20587**	**1.0000**
		0.025	no convergence		
Deterministic					
—				4.61410	—

Table 7.2.18: Result comparison for different values of n, p and ΔT.

7.2.6 Test 5: Goddard problem with one random variable

Moving on to a more complex version of the chance-constrained optimal control problem of the previous example, we now apply the KDE technique to the Goddard problem.

Formally, the structure of the model is the same as the previous example: the vertical ascent of a launcher in one dimension, in presence of a control $u(t) \in [0, 1]$ proportional to the thrust applied at time t. The main difference between Goddard problem and the one treated in Example 4 is the addition of the drag force to the dynamics. For the purpose of defining a probabilistic constraint, we consider the thrust T as the only random parameter and our objective is to maximize the final mass of the launcher while making sure that its altitude is higher than a given value r_f with a probability of at least p.

As in the previous example, a solution to this problem consists in an optimal control function $u^* : \mathbb{R}_+ \to [0, 1]$ such that, if we apply u^* regardless of the value of T, the probability of the final altitude being greater than r_f is greater than p.

Model

The original formulation of the Goddard problem can be found in [31]. We will consider a one-dimensional version of the one treated in [13].

The ODE system is

$$\begin{cases} \dot{r}(t) = v(t) & t \in (0, t_{\mathrm{f}}] \\ \dot{v}(t) = \frac{Tu(t) - Av^2(t)e^{-\kappa\left(r(t)-r_0\right)}}{m(t)} - \frac{1}{r^2(t)} & t \in (0, t_{\mathrm{f}}] \\ \dot{m}(t) = -bu(t) & t \in (0, t_{\mathrm{f}}] \\ r(0) = r_0 \\ v(0) = 0 \\ m(0) = m_0 \end{cases}$$

where the final time $t_{\mathrm{f}} > 0$ is free. The control function u belongs to $\mathcal{U}$, where

$$\mathcal{U} := \{u : \mathbb{R}_+ \to [0, 1] \subset \mathbb{R} \quad | \quad u \text{ is measurable}\}.$$

We will again integrate the equations numerically by using fourth-order Runge-Kutta method, as we did in the previous example, where the continuous control is replaced by a piece-wise constant function.

Before defining our stochastic optimization problem, we first show the solution to the deterministic one:

$$\begin{cases} \max\limits_{(t_{\mathrm{f}},u)\in\mathbb{R}_+\times\mathcal{U}} m(t_{\mathrm{f}}) \\ r(t_{\mathrm{f}}) \geq r_{\mathrm{f}} \end{cases} \tag{7.2.17}$$

which, implementing the numerical integration described above, is approximated by

$$\begin{cases} \max\limits_{(t_{\mathrm{f}},u)\in\mathbb{R}_+\times[0,1]^{n_t}} m(t_{\mathrm{f}}) \\ r(t_{\mathrm{f}}) \geq r_{\mathrm{f}}. \end{cases}$$

Table 7.2.19 sums up the choice of parameters for this model. We remark that all the quantities in this model are dimensionless and thus they do not require to be specified in terms of unit measures. The optimal cost found

Parameter	T	A	κ	b	r_0	m_0	r_{f}	n_t
Value	3.5	310	500	7	1	1	1.01	100

Table 7.2.19: Parameters for the deterministic optimization

by WORHP is

$$m(t_{\mathrm{f}}) \approx 0.62975.$$

Figures 7.2.22 shows the corresponding optimal trajectory.

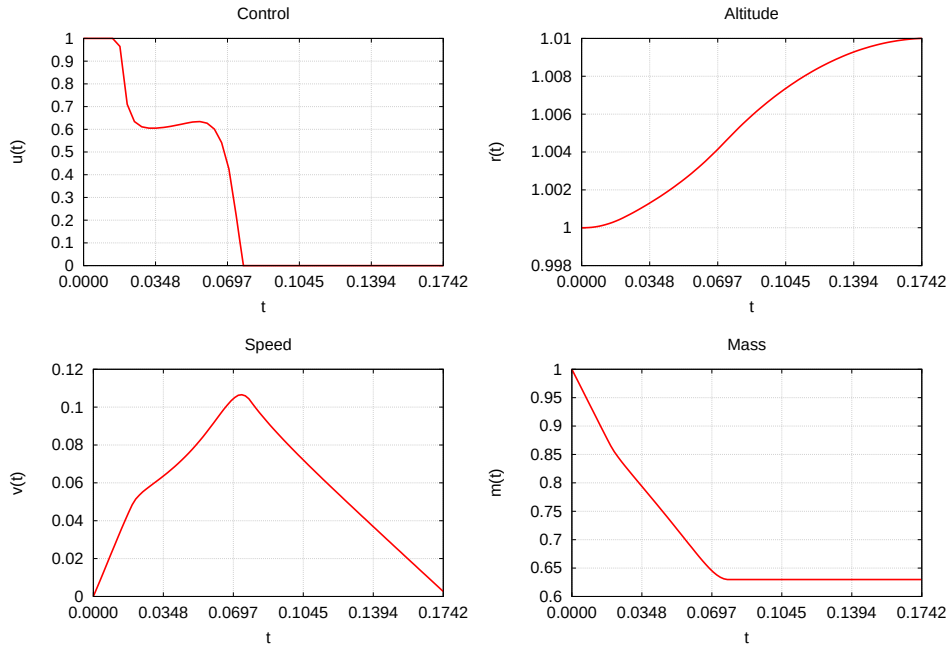


Figure 7.2.22: Plot of control, altitude, speed and mass for the Goddard problem.

Problem statement

Our goal to reach at least the altitude r_f with a 90% probability while maximizing the final mass of the launcher. Keeping in mind that the cost to be minimized also depends on the random parameter T, it has to be defined as an expectation.

$$\mathbb{E}\left[m(t_\mathrm{f})\right] = \mathbb{E}\left[\int_0^{t_\mathrm{f}} m_0 - \frac{T}{v_\mathrm{e}} u(t)dt\right] = \int_0^{t_\mathrm{f}} m_0 - \frac{\mathbb{E}[T]}{v_\mathrm{e}} u(t)dt.$$

We recall that T is a uniformly distributed random variable on the interval $[T_-, T_+]$ with expected value $\overline{T}$, so the cost is defined as

$$\overline{m}(t_\mathrm{f}) := \int_0^{t_\mathrm{f}} m_0 - \frac{\overline{T}}{v_\mathrm{e}} u(t)dt.$$

This leads us to the stochastic optimization problem

$$\begin{cases} \max\limits_{(t_\mathrm{f},u)\in\mathbb{R}_+\times\mathcal{U}} \overline{m}(t_\mathrm{f}) \\ \mathbb{P}\big[R_\mathrm{f}(T,t_\mathrm{f},u) \geq r_\mathrm{f}\big] \geq p \end{cases} \tag{7.2.18}$$

where $R_\mathrm{f}(T, t_\mathrm{f}, u)$ is the final altitude as a function of the random variable T, parameterized by u.

Table 7.2.20 shows the choice of parameters defined in this subsection.

Parameter	p	$\overline{T}$	ΔT
Value	0.9	3.5	0.1

Table 7.2.20: Additional parameters for the stochastic optimization

Solution via Kernel Density Estimation

By using the definition of the density function $f_{t_f,u}$ of the random variable $r_u(T, t_f)$, we can rewrite problem 7.2.18 as

$$\begin{cases} \max\limits_{(t_f,u)\in\mathbb{R}_+\times\mathcal{U}} \overline{m}(t_f) \\ F_{(t_f,u)}(r_f) \leq 1-p. \end{cases} \tag{7.2.19}$$

If we replace F_u with its KDE approximation $\hat{F}_u$, our problem becomes

$$\begin{cases} \max\limits_{(t_f,u)\in\mathbb{R}_+\times\mathcal{U}} \overline{m}(t_f) \\ \hat{F}_{(t_f,u)}(r_f) \leq 1-p. \end{cases} \tag{7.2.20}$$

The procedure used for solving problem (7.2.20) is described in the following steps.

1. **Draw the sample**
   Take n equidistant values for T in the interval $[T_-, T_+]$ such that

   $$T_i = T_- + (i-1)\frac{T_+ - T_-}{n-1} \quad \forall i \in 1, \dots, n.$$

2. **Define the constraint function**
   Instruct the solver on how to associate a pair of final time and control (t_f, u) to the constraint function $\hat{F}_{(t_f,u)}$.

   (a) For each T_i in $\{T_1, T_2, \dots, T_n\}$ define the elements of the sample $X(t_f, u)$ of R_f as $X_i(t_f, u) := R_f(T_i, t_f, u)$.

   (b) Build the KDE for the pdf of R_f using the SNR method (see 7.1.7) for computing the bandwidth.

   $$\hat{f}_{(t_f,u)}(x) := \frac{1}{nh}\sum_{i=1}^{n}\frac{1}{\sqrt{2\pi}}e^{-\frac{1}{2}\left(\frac{x-X_i(t_f,u)}{h}\right)^2}.$$

   (c) Compute $\hat{F}_{(t_f,u)}$ as

   $$\hat{F}_{(t_f,u)}(r_f) := \int_0^{r_f} \hat{f}_{(t_f,u)}(x)dx.$$

3. **Solve the problem**
   Now that the solver knows how to compute the approximation $\hat{F}_{(t_f,u)}$ of $F_{(t_f,u)}$, we can solve problem (7.2.16) as a standard deterministic optimization problem by using WORHP as described in Subsection 7.2.1 with an initial guess equal to the solution of the deterministic problem.

Results

Convergence of approximated solutions Figures 7.2.23 shows the behavior of the sequence of optimal costs for $n \in \{10, 20, 30, \ldots, 500\}$ and the corresponding rate of success $R := \frac{N_s}{N_a}$ computed a posteriori with $N_a = 10^5$. For example, for $n = 500$ the optimal cost is

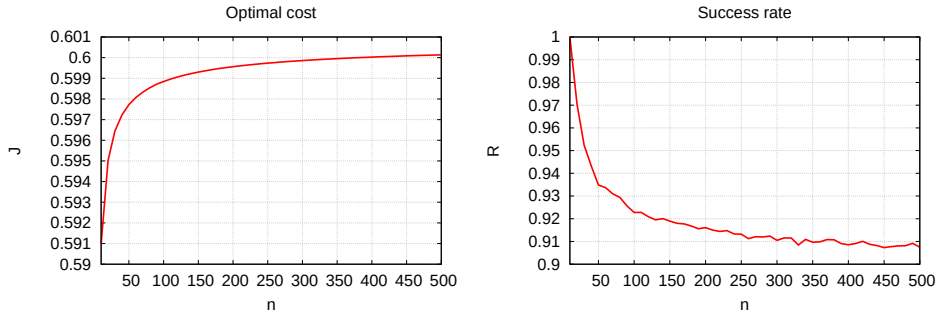


Figure 7.2.23: Plot of $m(t_f, u^*)$ and R as functions of n.

$$\overline{m}(t_f^*) \approx 0.60014$$

with a success rate $R = 90.81\%$. The corresponding optimal control u^* is shown in figure 7.2.24. Figure 7.2.25 the other related plots.

Comparison with best/worst case scenario Table 7.2.21 and Figure 7.2.26 compare the solution we just found for the stochastic optimization problem to the two solution we obtain from the deterministic one in the best and worst cases. It can be seen how the solution to the chance-

Case	T	t_f^*	$\overline{m}(t_f^*)$
Random	$\sim U(T_-, T_+)$	**0.18806**	**0.60014**
Best	T_+	**0.16126**	**0.65841**
Worst	T_-	**0.19016**	**0.59279**

Table 7.2.21: Result comparison for extremal values of T.

constrained problem is slightly better than the one in the worst case, but

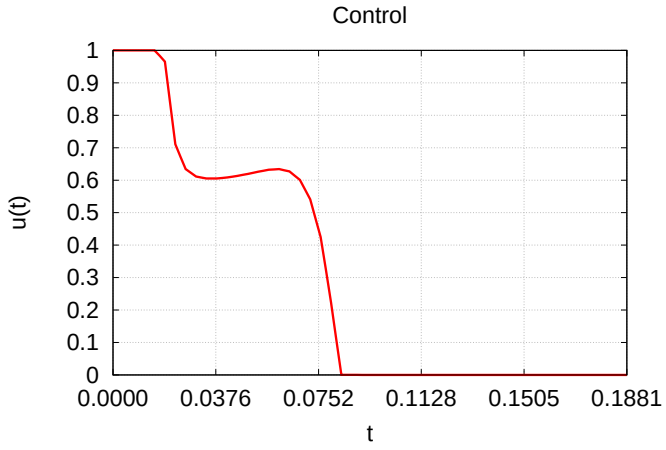


Figure 7.2.24: Optimal control for $n = 500$.

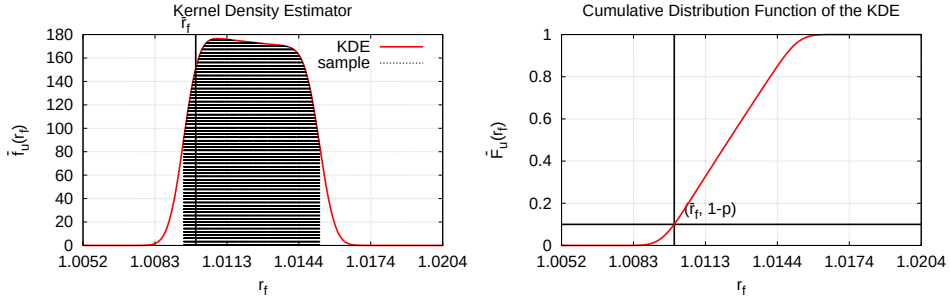


Figure 7.2.25: Plot of the Kernel Density Estimator $\hat{f}$ of $R_{\mathrm{f}}(T, t_{\mathrm{f}}^*, u^*)$ and its integral $\hat{F}$.

still lower than the one of the best case. Interestingly, Figure 7.2.26 shows that the shape of the control strategy doesn't change much between the three cases, and the main difference lies in the optimal value for the final time t_{f}^*.

Consistency Table 7.2.22 shows the comparison between the solution of the deterministic problem (7.2.17) and its stochastic counterpart (7.2.18) when p is close to 1 and T_i is close to 0. For the results in the table we set the initial guess for u equal to the optimal solution found for the deterministic problem (see Figure 7.2.22).

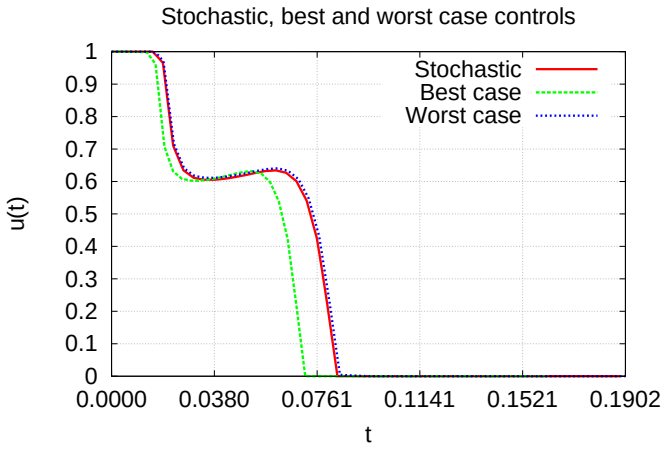


Figure 7.2.26: Comparison between stochastic, best and worst case controls.

n	p	ΔT	h	$\overline{m}(t_{\mathrm{f}}^*)$	R
Stochastic					
500	0.8	0.5	0.00517	**0.48082**	**0.7980**
		0.25	0.00155	**0.57006**	**0.8011**
		0.1	0.00048	**0.60852**	**0.7996**
		0.05	0.00022	**0.61970**	**0.7999**
		0.025	0.00010	**0.62669**	**0.6879**
	0.9	0.5	0.00813	**0.38665**	**0.9096**
		0.25	0.00185	**0.54187**	**0.9090**
		0.1	0.00051	**0.60014**	**0.9090**
		0.05	0.00023	**0.61625**	**0.8929**
		0.25	0.00011	**0.62221**	**0.9479**
	0.995	0.5	0.02127	**0.15383**	**1.0000**
		0.25	0.00271	**0.47279**	**1.0000**
		0.1	0.00057	**0.58270**	**1.0000**
		0.05	0.00024	**0.60750**	**1.0000**
		0.025	0.00011	**0.61860**	**1.0000**
Deterministic					
—				0.62975	—

Table 7.2.22: Result comparison for different values of n, p and ΔT.

7.2.7 Test 6: Complex three stage launcher with one decision variable and two random variables

We now move to a different application of this method. We use the more complex model of a real space launcher for defining a percentile optimization problem in the form of (6.1.12). In this case we have two random parameters: the specific impulse I_{sp_3} and the index K_3 of the third stage. As a function of both I_{sp_3} and K_3 the optimal fuel mass of the third stage is also random, and our goal is to compute the 0.9-percentile of its distribution.

Model

This subsection is aimed at describing the mathematical model which represents the dynamics of the launcher: after defining the coordinate system related to the chosen frame of reference, we list all the forces that the vehicle is subject to and then write the system's equations of motion.

Frame of reference We define the inertial equatorial frame coordinate system $\mathcal{S} := (O, \mathbf{i}, \mathbf{j}, \mathbf{k})$, where

- O is the center of the Earth;
- $\mathbf{k}$ is the versor of Earth rotation axis, directed towards North;
- $\mathbf{i}$ is the versor that belongs to Earth equatorial plane and points towards the Greenwich meridian;
- $\mathbf{j} := \mathbf{k} \times \mathbf{i}$ completes the coordinate system.

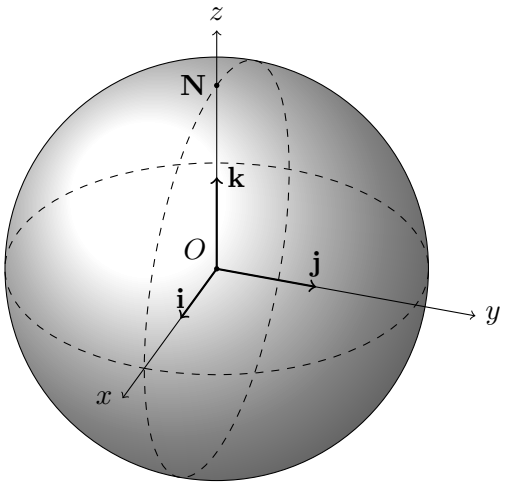


Figure 7.2.27: The coordinate system $\mathcal{S}$.

In this coordinate system we define

$$\begin{aligned} \mathbf{x} &:= x\mathbf{i} + y\mathbf{j} + z\mathbf{k} \\ \mathbf{v} &:= \dot{\mathbf{x}} := v_x\mathbf{i} + v_y\mathbf{j} + v_z\mathbf{k} \\ \mathbf{v}_{\mathrm{r}}(\mathbf{v}, \mathbf{x}) &:= \mathbf{v} - (0, 0, \Omega) \times \mathbf{x} \end{aligned}$$

to be respectively the position, the velocity and the relative velocity of the vehicle's center of mass G, where Ω is the Earth's angular speed.
Furthermore, we will denote with (ϕ, λ, h) the geographic coordinates of G, as shown in figure 7.2.28.

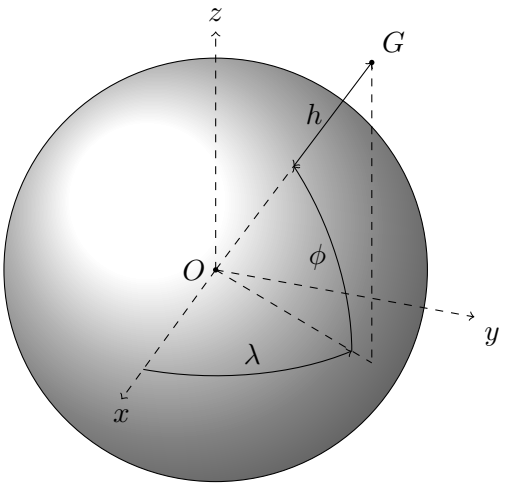


Figure 7.2.28: The geographic coordinates of G.

Cartesian to geographic coordinates transformation At a given time t, the equivalence between cartesian and geographic coordinates is given by the following relations (see [30] for more details).

- Latitude

$$\begin{cases} \frac{p}{\cos(\phi)} - \frac{z}{\sin(\phi)} - e^2\nu(\phi) = 0 & p \neq 0 \\ \phi = \frac{\pi}{2} & p = 0 \text{ and } z \geq 0 \\ \phi = -\frac{\pi}{2} & p = 0 \text{ and } z < 0 \end{cases}$$

  where

$$\begin{aligned} e &:= \sqrt{1 - \frac{R_{\mathrm{p}}^2}{R_{\mathrm{e}}^2}} \\ \nu(\phi) &:= \frac{R_{\mathrm{e}}}{\sqrt{1 - e^2\sin^2(\phi)}} \\ p &:= \sqrt{x^2 + y^2} \end{aligned}$$

R_e and R_p are the Earth's equatorial and polar radius.

- Longitude

$$\lambda = \begin{cases} \arccos\left(\frac{x}{p}\right) - \Omega t & p \neq 0 \text{ and } y \geq 0 \\ -\arccos\left(\frac{x}{p}\right) - \Omega t & p \neq 0 \text{ and } y < 0 \\ 0 & p = 0. \end{cases}$$

- Height

$$\begin{cases} h = \frac{p}{\cos(\phi)} - \nu(\phi) & p \neq 0 \\ h = |z| - R_p & p = 0. \end{cases}$$

Geographic to cartesian coordinates transformation At a given time t, the following equations show how to change from geographic to cartesian coordinates.

$$\begin{aligned} x &= (\nu(\phi) + h)\cos(\phi)\cos(\lambda + \Omega t) \\ y &= (\nu(\phi) + h)\cos(\phi)\sin(\lambda + \Omega t) \\ z &= \left(\nu(\phi)(1 - e^2) + h\right)\sin(\phi). \end{aligned}$$

Orbit plane The orbit plane is the plane of the ellipse that defines the GTO, it is characterized by two angles: the longitude of the ascending node and the angle of inclination with respect to the equatorial plane of the Earth. Not all the inclinations can be reached from a given launch site: the location has to be a point inside the target orbit plane.

Axis and angles We associate to the launcher a longitudinal axis: this axis passes through G and points towards the edge of the launcher (see figure 7.2.29). At each time the thrust of the launcher has the same direction of the longitudinal axis (i.e. we are assuming a perfect control).
We also define the following angles:

- The launch azimuth ψ is the angle between the perpendicular line to the longitudinal axis at the initial position directed towards North and the orbit plane. The launch azimuth must satisfy the following equation in order to allow the launcher to reach the target orbit inclination

$$\psi = \arcsin\left(\frac{\cos(i)}{\cos(\phi_0)}\right)$$

this means that the inclination i must be greater than the launch site latitude ϕ_0.

- The angle of attack α between the longitudinal axis and the relative velocity $\mathbf{v}_\mathrm{r}$ measured in the orbit plane;
- The pitch angle θ between the longitudinal axis and the vector $\overrightarrow{O\mathbf{x}_0}$ measured in the orbit plane;

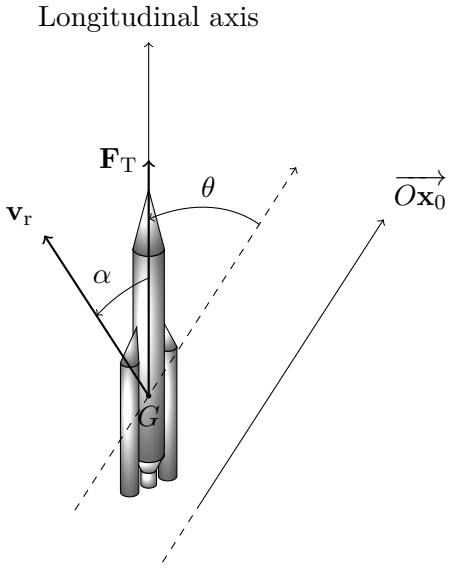


Figure 7.2.29: The angles θ and α.

Mechanical and structural parameters We call β_i, I_{sp_i} and S_i respectively the mass flow rate, the specific impulse and the area of the nozzle's section of the i-th stage engine. Furthermore, we denote with A_i the area of the i-th stage reference surface involved in the computation of the drag force. Finally, we call m the total mass of the vehicle. Depending on the flight phase, it is the sum of some of the payload m_p, payload case m_c, the fairing m_f, the i-th stage fuel $m_{\mathrm{e}i}(t)$ at time t, where the initial fuel mass of each stage is defined as

$$m_{\mathrm{e}i0} := m_{\mathrm{e}i}(t_0) \quad \forall i \in \{1, 2, 3\}$$

the i-th stage structure $m_{\mathrm{s}i}$, which is defined as

$$m_{\mathrm{s}i} := K_i m_{\mathrm{e}i0}$$

with K_i being the i-th stage index.

We now define the mathematical model of the dynamic system: we first introduce all the forces that we choose to take into account, then we describe its evolution through Newton's second law.

Force of gravity Gravity is given by

$$\mathbf{F}_{\mathrm{G}}(m,\mathbf{x}) = -\begin{pmatrix} F_{\mathrm{G}x}(m,\mathbf{x}) & 0 & 0 \\ 0 & F_{\mathrm{G}y}(m,\mathbf{x}) & 0 \\ 0 & 0 & F_{\mathrm{G}z}(m,\mathbf{x}) \end{pmatrix} \frac{\mathbf{x}}{||\mathbf{x}||}$$

where

$$F_{\mathrm{G}x}(m,\mathbf{x}) = F_{\mathrm{G}y}(m,\mathbf{x}) = m\frac{\mu_0}{||\mathbf{x}||^2}\left(1 + J_2\frac{3}{2}\frac{R_{\mathrm{e}}^2}{||\mathbf{x}||^2}\left(1 - 5\frac{z^2}{||\mathbf{x}||^2}\right)\right)$$
$$F_{\mathrm{G}z}(m,\mathbf{x}) = m\frac{\mu_0}{||\mathbf{x}||^2}\left(1 + J_2\frac{3}{2}\frac{R_{\mathrm{e}}^2}{||\mathbf{x}||^2}\left(3 - 5\frac{z^2}{||\mathbf{x}||^2}\right)\right)$$

μ_0 is the standard gravitational parameter of the Earth and J_2 is the correction factor due to its oblateness.

Drag force Given by

$$\mathbf{F}_{\mathrm{D}}(\mathbf{x},\mathbf{v}) = -F_{\mathrm{D}}(\mathbf{x},\mathbf{v})\frac{\mathbf{v}_{\mathrm{r}}(\mathbf{x},\mathbf{v})}{||\mathbf{v}_{\mathrm{r}}(\mathbf{x},\mathbf{v})||}$$

where

$$F_{\mathrm{D}}(\mathbf{x},\mathbf{v}) = \frac{1}{2}\rho(\mathbf{x})||\mathbf{v}_{\mathrm{r}}(\mathbf{x},\mathbf{v})||^2 A C_{\mathrm{D}}(\mathbf{x},\mathbf{v})$$

ρ is the air density and C_{D} is the drag coefficient, depending on the Mach number

$$M_{\mathrm{a}}(\mathbf{x},\mathbf{v}) = \frac{||\mathbf{v}_{\mathrm{r}}(\mathbf{x},\mathbf{v})||}{v_{\mathrm{s}}(\mathbf{x})}$$

which itself depends on the speed of sound v_{s}.

Thrust force

$$\mathbf{F}_{\mathrm{T}}(\theta,\mathbf{x},\mathbf{v}) = F_{\mathrm{T}}(\mathbf{x})\mathbf{i}_{\mathrm{T}}(\theta,\mathbf{x},\mathbf{v})$$

where

$$F_{\mathrm{T}}(\mathbf{x}) = g_0\beta I_{\mathrm{sp}} - SP(\mathbf{x})$$

g_0 is the Earth gravitational acceleration and P is the atmospheric pressure. The direction $\mathbf{i}_{\mathrm{T}}$ is given by

$$\mathbf{i}_{\mathrm{T}}(\theta,\mathbf{x},\mathbf{v}) = \begin{cases} \frac{\mathbf{v}_{\mathrm{r}}(\mathbf{x},\mathbf{v})}{||\mathbf{v}\mathrm{r}(\mathbf{x},\mathbf{v})||} & \alpha = 0 \\ \mathbf{R}_{\lambda_0,\phi_0}\mathbf{R}_{\psi}\mathbf{R}(\theta)\mathbf{e}_1 & \alpha \neq 0 \end{cases}$$

where

$$\begin{aligned}
\mathbf{R}_{\lambda_0,\phi_0} &= \begin{pmatrix} -\sin(\lambda_0) & -\cos(\lambda_0)\sin(\phi_0) & \cos(\lambda_0)\cos(\phi_0) \\ \cos(\lambda_0) & -\sin(\lambda_0)\sin(\phi_0) & \sin(\lambda_0)\cos(\phi_0) \\ 0 & \cos(\phi_0) & \sin(\phi_0) \end{pmatrix} \\
\mathbf{R}_{\psi} &= \begin{pmatrix} 0 & \sin(\psi) & -\cos(\psi) \\ 0 & \cos(\psi) & \sin(\psi) \\ 1 & 0 & 0 \end{pmatrix} \\
\mathbf{R}(\theta) &= \begin{pmatrix} \cos(\theta) & -\sin(\theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{pmatrix} \\
\mathbf{e}_1 &= (1,0,0)^\intercal
\end{aligned}$$

λ_0 and ϕ_0 are the longitude and the latitude of the launch site and ψ is the launch azimuth.

Equations of motion We can now write the equations of motion in cartesian coordinates:

$$\begin{cases} \dot{\mathbf{x}}(t) = \mathbf{v}(t) \\ m(t)\dot{\mathbf{v}}(t) = \mathbf{F}_{\mathrm{G}}(m(t), \mathbf{x}(t)) + \mathbf{F}_{\mathrm{D}}(\mathbf{x}(t), \mathbf{v}(t)) + \mathbf{F}_{\mathrm{T}}(\theta(t), \mathbf{x}(t), \mathbf{v}(t)) \\ \dot{m}(t) = -\beta. \end{cases} \tag{7.2.21}$$

We can control the direction of the launcher by acting on the pitch angle θ at any time t.

Target orbit For a given position $\mathbf{x}$ and velocity $\mathbf{v}$, the perigee and apogee of the orbit associated are given by

$$\begin{aligned}
L_{\mathrm{p}}(\mathbf{x}, \mathbf{v}) &= (1 - \epsilon(\mathbf{x}, \mathbf{v}))a(\mathbf{x}, \mathbf{v}) - R_{\mathrm{e}} \\
L_{\mathrm{a}}(\mathbf{x}, \mathbf{v}) &= (1 + \epsilon(\mathbf{x}, \mathbf{v}))a(\mathbf{x}, \mathbf{v}) - R_{\mathrm{e}}
\end{aligned}$$

where ϵ is the eccentricity of the orbit

$$\epsilon(\mathbf{x}, \mathbf{v}) = \sqrt{1 - \frac{||\mathbf{x} \times \mathbf{v}||^2}{\mu_0 a(\mathbf{x}, \mathbf{v})}}$$

and a is the semi-major axis

$$a(\mathbf{x}, \mathbf{v}) = \frac{1}{\frac{2}{||\mathbf{x}||} - \frac{||\mathbf{v}||^2}{\mu_0}}.$$

Flight sequence The flight sequence consists in several phases, we will use the following notation to denote duration and final time of each flight phase: t_0 is the initial time, τ_i is the duration of the phase i, $\tau_{i.j}$ is the duration of the sub-phase $i.j$, t_i is the final time of the phase i and $t_{i.j}$ is the final time of the sub-phase $i.j$.

Phase 1 The launch azimuth is fixed at the value ψ and the initial position at the geographic coordinates (ϕ_0, λ_0, h_0). During this phase the mass of the launcher is

$$m(t) = m_{\mathrm{p}} + m_{\mathrm{c}} + m_{\mathrm{f}} + \sum_{i=1}^{3}(1 + K_i)m_{\mathrm{e}i}(t) \quad \forall t \in [t_0, t_1).$$

1.1 The engine of the first stage is ignited and the launcher accelerates vertically (i.e. with the same direction of $\overrightarrow{OG}$) leaving the service structure

$$\theta(t) \equiv 0 \quad \forall t \in [t_0, t_{1.1}).$$

1.2 The launcher rotates with constant speed changing its orientation:

$$\theta(t) = \frac{\theta_1}{\tau_{1.2}}(t - t_{1.1}) \quad \forall t \in [t_{1.1}, t_{1.2}).$$

1.3 The direction of the thrust is fixed at the final values of the previous sub-phase until the angle of incidence α is zero (see figure 7.2.29):

$$\begin{aligned} \theta(t) &= \theta_1 \quad \forall t \in [t_{1.2}, t_{1.3}) \\ t_{1.3} &= \min_{t \in (t_{1.2}, +\infty)} \{t \quad | \quad \alpha(t) = 0\}. \end{aligned}$$

1.4 Zero incidence flight until complete consumption of the first stage fuel:

$$\tau_1 = \frac{m_{\mathrm{e}10}}{\beta_1}$$

this sub-phase ends with the separation of the first stage.

Phase 2 At the beginning of this phase the mass of the launcher is

$$m(t) = m_{\mathrm{p}} + m_{\mathrm{c}} + m_{\mathrm{f}} + \sum_{i=2}^{3}(1 + K_i)m_{\mathrm{e}i}(t) \quad \forall t \in [t_1, t_{2.1}).$$

2.1 Ignition of second stage engine. This sub-phase ends with the release of the fairing, as soon as the heat flux decreases to a given value:

$$\begin{aligned} \theta(t) &= \theta_2 + \theta_2'(t - t_1) \quad \forall t \in [t_1, t_{2.1}) \\ t_{2.1} &= \min_{t \in (t_1, +\infty)} \left\{t \quad | \quad \Gamma\big(\mathbf{x}(t), \mathbf{v}(t)\big) \leq \Gamma^*\right\} \end{aligned}$$

where

$$\Gamma(\mathbf{x}, \mathbf{v}) = \frac{1}{2}\rho(\mathbf{x})||\mathbf{v}_r(\mathbf{x}, \mathbf{v})||^3$$

represents the heat flux.
The mass changes to

$$m(t) = m_{\mathrm{p}} + m_{\mathrm{c}} + \sum_{i=2}^{3}(1 + K_i)m_{\mathrm{e}i}(t) \quad \forall t \in [t_{2.1}, t_2).$$

2.2 Flight without fairing until complete consumption of the fuel in the second stage:

$$\tau_2 = \frac{m_{\mathrm{e}20}}{\beta_2}$$

this sub-phase ends with the jettison of the second stage:

$$\theta(t) = \theta_2 + \theta_2'\tau_{2.1} + \theta_2'(t - t_1) \quad \forall t \in [t_{2.1}, t_2).$$

Phase 3 During this phase the mass of the launcher is

$$m(t) = m_{\mathrm{p}} + m_{\mathrm{c}} + (1 + K_3)m_{\mathrm{e}3}(t) \quad \forall t \in [t_{2.2}, t_{\mathrm{f}})$$

Ignition of third stage engine, this phase ends when the third stage's fuel is exhausted:

$$\tau_3 = \frac{m_{\mathrm{e}30}}{\beta_3}.$$

At final time $t_{\mathrm{f}} := t_3$ the the final position and velocity have to be compatible with the target orbit:

$$\begin{aligned} \theta(t) &= \theta_3 + \theta_3'(t - t_2) \quad \forall t \in [t_{2.2}, t_{\mathrm{f}}) \\ L_{\mathrm{p}}(\mathbf{x}(t_{\mathrm{f}}), \mathbf{v}(t_{\mathrm{f}})) &= L_{\mathrm{p}}^* \\ L_{\mathrm{a}}(\mathbf{x}(t_{\mathrm{f}}), \mathbf{v}(t_{\mathrm{f}})) &= L_{\mathrm{a}}^*. \end{aligned}$$

Optimization of the third stage mass We can now formulate the following deterministic optimization problem.

$$\begin{cases} \min\limits_{(m_{\mathrm{e}30}, \theta_1, \theta_2, \theta_2', \theta_3, \theta_3') \in \mathbb{R}_+^6} m_{\mathrm{e}30} \\ L_{\mathrm{p}}(m_{\mathrm{e}30}, \theta_1, \theta_2, \theta_2', \theta_3, \theta_3') = L_{\mathrm{p}}^* \\ L_{\mathrm{a}}(m_{\mathrm{e}30}, \theta_1, \theta_2, \theta_2', \theta_3, \theta_3') = L_{\mathrm{a}}^* \end{cases} \tag{7.2.22}$$

where, with a slight abuse of notation, the functions L_{p} and L_{a} denote, respectively, the perigee and apogee associated to the final state $(\mathbf{x}(t_{\mathrm{f}}), \mathbf{v}(t_{\mathrm{f}}))$, according to the decision variables $(m_{\mathrm{e}30}, \theta_1, \theta_2, \theta_2', \theta_3, \theta_3')$.

Fairing	
m_f	1.100 kg

Case	
m_c	858.86 kg

Payload	
m_p	4500 kg

Stage 1	
K_1	0.13
β_1	1896.58 kg/s
$I_{\mathrm{sp}1}$	345.32 s
S_1	7.18 m^2
A_1	17.35 m^2

Stage 2	
K_2	0.13
β_2	273.49 kg/s
$I_{\mathrm{sp}2}$	349.4 s
S_2	5.16 m^2
A_2	17.35 m^2

Stage 3	
K_3	0.13
β_3	42.18 kg/s
$I_{\mathrm{sp}3}$	450.72 s
S_3	1.97 m^2
A_3	17.35 m^2

Table 7.2.23: Mechanical and structural parameters

Given data and fixed parameters Table 7.2.23 summarizes the choice of all the fixed parameters of the problem while Figure 7.2.30 shows the profile of the speed of sound, the air density, the atmospheric pressure (each one depending on altitude) and drag coefficient (depending on the Mach number). With this choice of the duration of the first two flight phases, the fuel load of the corresponding stages can be computed easily (see Table 7.2.26) because of the relation $m_{ei0} = \beta_i \tau_i$ for $i \in \{1, 2, 3\}$. Lastly, parameter values for the Earth and the flight sequence are shown in Tables 7.2.24 and 7.2.25 respectively.

Ω	$7.292155{\cdot}10^{-5}$ rad/s
R_p	6356752 m
R_e	6378137 m
μ_0	$3.986005{\cdot}10^{14}$ $\mathrm{m}^3/\mathrm{s}^2$
J_2	$1.08263{\cdot}10^{-3}$
g_0	9.80665 m/s^2

Table 7.2.24: Earth's parameters

The optimal solution In order to use WORHP to obtain the numerical solution to (7.2.22), we have to rewrite our optimization problem in the form

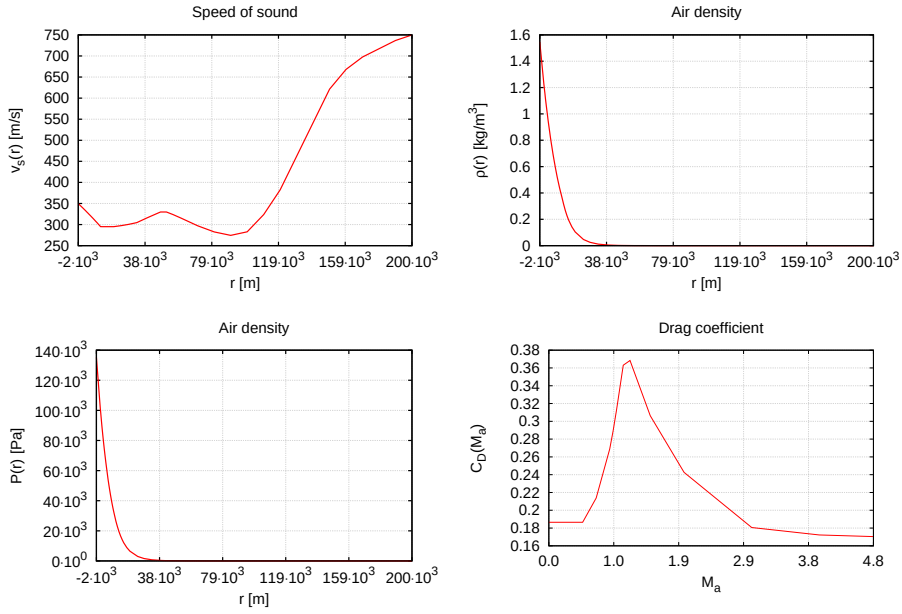


Figure 7.2.30: Speed of sound v_s, air density ρ, atmospheric pressure P and drag coefficient C_D.

(7.2.1). This can be done quite easily by setting

$$
\begin{aligned}
N &:= 6 \\
X &:= (m_{\mathrm{e}30}, \theta_1, \theta_2, \theta_2', \theta_3, \theta_3')^\mathsf{T} \\
X_\mathrm{L} &:= (0, 0, 0, 0, 0, 0)^\mathsf{T} \\
X_\mathrm{U} &:= (10000 \text{ kg}, 5 \text{ deg}, 180 \text{ deg}, 0.5 \text{ deg/s}, 180 \text{ deg}, 0.5 \text{ deg/s})^\mathsf{T} \\
F(X) &:= m_{\mathrm{e}30} \\
M &:= 2 \\
G(X) &:= (L_\mathrm{p}(X), L_\mathrm{a}(X))^\mathsf{T} \\
G_\mathrm{L} &:= (L_\mathrm{p}^*, L_\mathrm{a}^*)^\mathsf{T} \\
G_\mathrm{U} &:= G_\mathrm{L}.
\end{aligned}
$$

The ODE system (7.2.21) is integrated by using the Fortran 90 subroutine DOP853 described in [32]. The optimal values found by WORHP for the optimization variables are reported in Table 7.2.27 and Figure 7.2.31 shows the corresponding optimal trajectory.

Problem statement

Let $M_{\mathrm{e}30}(\pi, m_\mathrm{p})$ be the value function of problem (7.2.22), depending on $\pi := (I_{\mathrm{sp}_3}, K_3)$ and the dimensioning parameter m_p. Consider the following

Phase 1	Sub-phase 1.1	t_0	0 s
		ψ	90 deg
		ϕ_0	5.159722 deg
		λ_0	-52.650278 deg
		h_0	0 m
		$\tau_{1.1}$	5 s
	Sub-phase 1.2	$\tau_{1.2}$	2 s
		τ_1	147 s
Phase 2	Sub-phase 2.1	Γ^*	1135 W/m^2
		τ_2	222 s
Phase 3		L_{p}^*	200000 m
		L_{a}^*	35786000 m

Table 7.2.25: Parameters for the flight sequence

$m_{\mathrm{e}10}$	278797.26 kg
$m_{\mathrm{e}20}$	60714.78 kg

Table 7.2.26: Values for the initial fuel masses

chance-constrained optimization problem

$$\begin{cases} \min\limits_{\mu \in \mathbb{R}_+} \mu \\ \mathbb{P}[M_{\mathrm{e}30}(\pi, m_{\mathrm{p}}) \geq \mu] \geq p \end{cases} \tag{7.2.23}$$

where I_{sp_3} and K_3 are uniformly distributed random variables, respectively on the intervals $[I_{\mathrm{sp}_{3-}}, I_{\mathrm{sp}_{3+}}]$ and $[K_{3-}, K_{3+}]$, with expected values $\overline{I}_{\mathrm{sp}_3}$ and $\overline{K}_3$:

$$I_{\mathrm{sp}_3} \sim U(I_{\mathrm{sp}_{3-}}, I_{\mathrm{sp}_{3+}})$$
$$K_3 \sim U(K_{3-}, K_{3+}).$$

Here $I_{\mathrm{sp}_{3-}} := \overline{I}_{\mathrm{sp}_3}(1 - \Delta I_{\mathrm{sp}_3})$, $I_{\mathrm{sp}_{3+}} := \overline{I}_{\mathrm{sp}_3}(1 + \Delta I_{\mathrm{sp}_3})$ (similar definitions hold for K_3). Note that (7.2.23) matches the definition of the percentile optimization problem (6.1.12).

We remark that this problem, and thus its solution, depends on two dimensioning parameters: the payload m_{p} and the success probability p.

Table 7.2.28 shows the choice of parameters defined in this subsection.

The main difference between this problem and the ones treated previously is that the decision variable is separated from the random ones. More generally, if we call x and ξ respectively the decision and the random variables, we can rewrite the chance constraint in the general form

$$\mathbb{P}[G(x, \xi) \leq 0] \geq p.$$

m_{e30}	2627.1511 kg
θ_1	1.98164037 deg
θ_2	74.24468871 deg
θ_2'	0.14736836 deg/s
θ_3	99.15421943 deg
θ_3'	0.30801744 deg/s

Table 7.2.27: Optimal values for the free variables

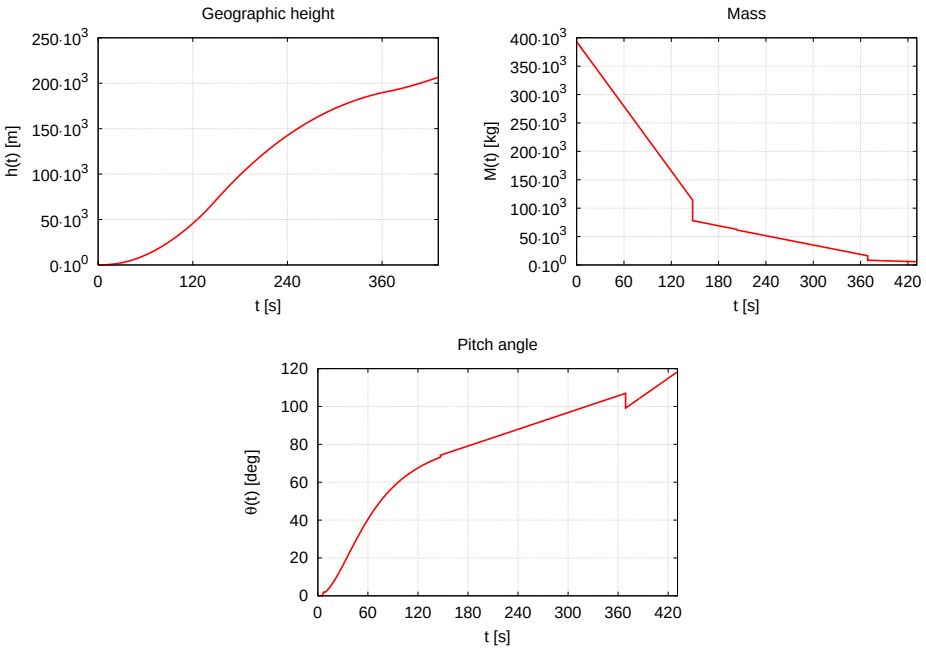


Figure 7.2.31: Result of the three-stage launcher optimization

In the particular case of this example's model, the inequality above can be rewritten as

$$\mathbb{P}[D(\pi) \leq E(\mu)] \geq p$$

allowing us to improve the solver's performances by pre-computing the function $M_{e30}(\pi, m_p)$ at given grid values for the random variables π for a fixed m_p. In the opposite case in which x and ξ are not separated, we would need to compute the constraint function also for all the possible values of x, which can be unbounded. Figure 7.2.32 shows the plot of $M_{e30}(\pi, m_p)$ as a function of π for our choice of m_p (see Table 7.2.23). The function has been evaluated at 16 values of π on an equally partitioned grid on the set $[I_{sp3-}, I_{sp3+}] \times [K_{3-}, K_{3+}]$. The values in between gridpoints are obtained via bilinear interpolation. We also recall that, since the constraint function is parameterized by the payload m_p, every change in its value would re-

Parameter	p	$\overline{I}_{\mathrm{sp}_3}$	ΔI_{sp_3}	$\overline{K}_3$	ΔK_3
Value	0.9	450.72 [s]	0.1	0.13	0.1

Table 7.2.28: Additional parameters for the stochastic optimization

quire a new computation of $M_{\mathrm{e}30}$ at grid values. For all the values of π in

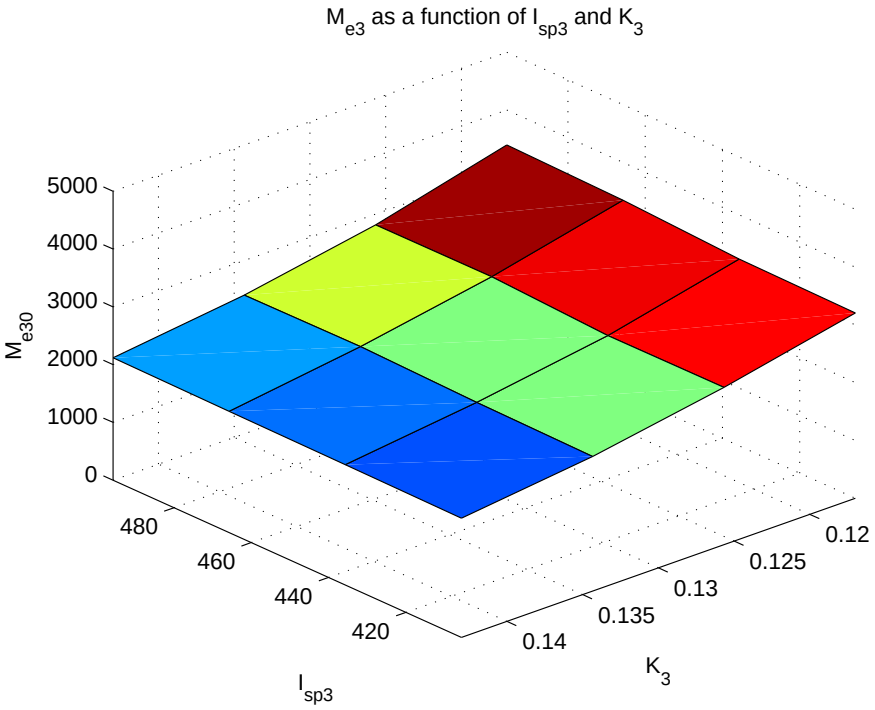


Figure 7.2.32: Plot of the third stage optimal fuel mass as a function of π.

$[I_{\mathrm{sp}3-}, I_{\mathrm{sp}3+}] \times [K_{3-}, K_{3+}]$ the solver WORHP was able to compute an optimal control allowing the launcher to reach its final orbit while minimizing the initial mass.

Solution via Kernel Density Estimation

In order to use the KDE we have to reformulate the chance constraint showing its dependency on the CDF F of random variable $M_{\mathrm{e}30}(\pi, m_{\mathrm{p}})$. Let $f_{m_{\mathrm{p}}}$ be the pdf of $M_{\mathrm{e}30}$, parameterized by m_{p}. From the definition of $f_{m_{\mathrm{p}}}$ we can rewrite problem 7.2.23 as

$$\begin{cases} \min\limits_{\mu \in \mathbb{R}_+} \mu \\ F_{m_{\mathrm{p}}}(\mu) \geq 1 - p. \end{cases} \tag{7.2.24}$$

As explained earlier, the remarkable feature of the problem is that, in contrast with the previous examples, the PDF estimator does not depend on the optimization parameter μ.

For each choice of m_p we are able to produce an approximation $\hat{F}_{m_\mathrm{p}}$ of F_{m_p} via KDE by drawing a sample of size from the array of random variables π. Our problem becomes

$$\begin{cases} \min\limits_{\mu \in \mathbb{R}_+} \mu \\ \hat{F}_{m_\mathrm{p}}(\mu) \geq 1 - p. \end{cases} \tag{7.2.25}$$

The procedure used for solving problem (7.2.25) is described in the following steps.

1. **Draw the sample**
   We take n random realizations of π according to the distribution of its elements. This sample can be represented as a 2 by n matrix:
   $$\begin{pmatrix} I_{\mathrm{sp}31} & I_{\mathrm{sp}32} & \dots & I_{\mathrm{sp}3n} \\ K_{31} & K_{32} & \dots & K_{3n}. \end{pmatrix}$$

2. **Define the constraint function**
   Instruct the solver on how to associate μ to the constraint function $\hat{F}_{m_\mathrm{p}}(\mu)$.

   (a) For performance purposes, the best strategy would be to sample the function $M_{\mathrm{e}30}(\pi, m_\mathrm{p})$ and store its values in a file to be used when building the estimator for the distribution function $F_{m_\mathrm{p}}(\mu)$. This will avoid nested optimization procedures and give us the possibility to solve in advance convergence issues with the deterministic optimization problem. For each element π_i of the sample set $X_i(m_p) := M_{\mathrm{e}30}(\pi_i, m_\mathrm{p})$.

   (b) Build the KDE for the pdf of $M_{\mathrm{e}30}$ using the SNR method (see 7.1.7) for computing the bandwidth.
   $$\hat{f}_{m_\mathrm{p}}(x) := \frac{1}{nh} \sum_{i=1}^{n} \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x - X_i(m_\mathrm{p})}{h}\right)^2}.$$

   (c) Compute $\hat{F}_{m_\mathrm{p}}$ as
   $$\hat{F}_{m_\mathrm{p}}(\mu) := \int_0^\mu \hat{f}_{m_\mathrm{p}}(x) dx.$$

3. **Solve the problem**
   Now that the solver knows how to compute the approximation $\hat{F}_{m_\mathrm{p}}$ of F_{m_p}, we can solve problem (7.2.25) as a standard deterministic optimization problem by using WORHP as described in Subsection 7.2.1 with an initial guess equal to the solution of the deterministic problem.

Results

Convergence of approximated solutions Figures 7.2.33 to 7.2.34 show the behavior of ten sequences of optimal costs for $n \in \{10, 20, 30, \dots, 500\}$ and the corresponding rate of success $R := \frac{N_s}{N_a}$ computed a posteriori with $N_a = 10^5$. For example, for $n = 500$ the optimal cost and the success rate

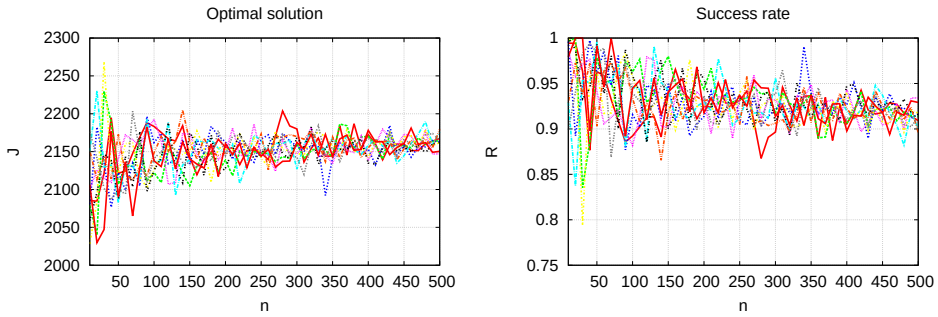


Figure 7.2.33: Plot of μ^* and R as functions of n (ten simulations).

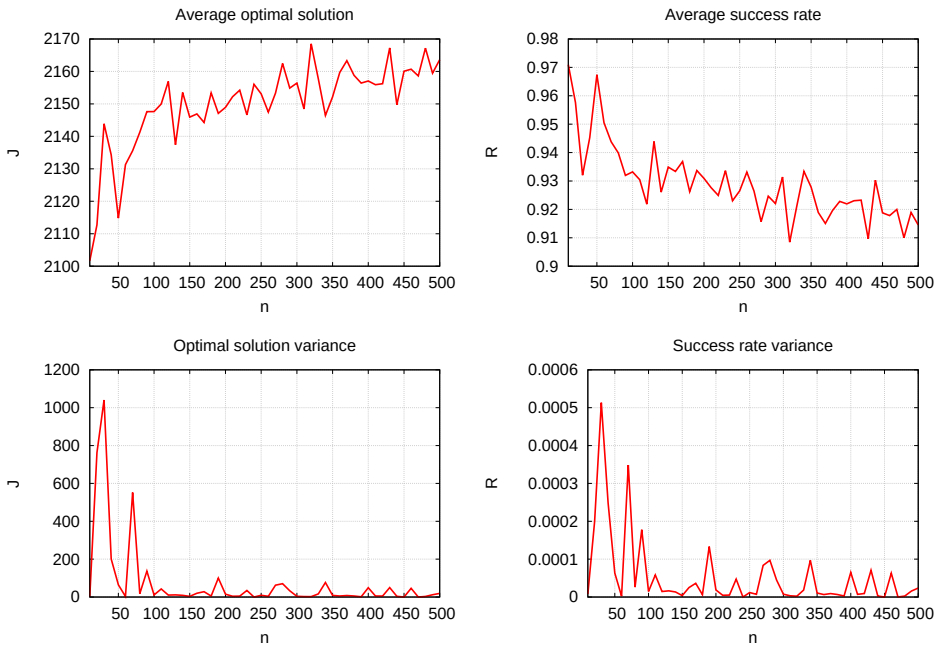


Figure 7.2.34: Plot of the average value and variance of μ^* and R as functions of n.

are

$$\mu^* \approx 2162.78$$

and $R \approx 91.83\%$. Figure 7.2.35 shows the related plots.

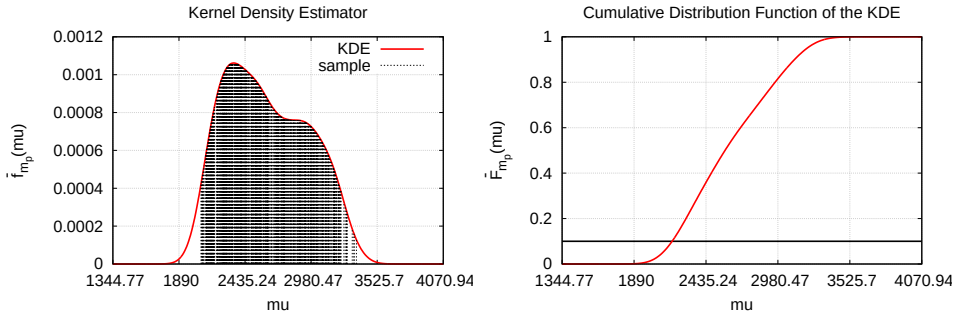


Figure 7.2.35: Plot of the Kernel Density Estimator $\hat{f}_{m_\mathrm{p}}$ of $M_{\mathrm{e}30}$ and its integral $\hat{F}$.

Chapter 8

Conclusions

Throughout this part of the thesis we showed how chance-constrained optimization can be a valid approach for solving robust optimization and optimal control problems, especially when the traditional deterministic techniques like the worst-case analysis cannot be applied since they are not designed to take into account unfeasible solutions. Although we did not focus on the generalization of the existing theoretical results (e.g. to the case where decision and random variables are not separated) we recognize that such a study would be fundamental to the development of chance-constrained optimization.

Nonetheless, even in lack of a solid theoretical framework, the numerical results obtained with Kernel Density Estimation are very promising. We applied this technique to increasingly complex optimization problems with chance constraints and it always led to satisfying solutions, while still offering room for improvement.

Better results might be obtained by changing the computation of the bandwidth h, for example, by substituting the second derivative f'' of the unknown density in (7.1.6) with some approximation (this so called *plug-in* method is explained in detail in [65]). Such a method can improve the accuracy of the estimator $\hat{f}$, but since it involves more complex operations for the computation of h compared to the Simple Normal Reference bandwidth (7.1.7), we decided to implement the latter in our tests in order to preserve performances. Another way for improving the accuracy of the estimator $\hat{f}$ is to experiment with different kernels. This, however, should not be the priority because the Gaussian kernel satisfies all the required regularity properties. The regularity of this kernel, coupled with the recurrence of the corresponding Gaussian distribution in the real world, makes it a widely popular choice in the literature, which is usually focused on the study of bandwidth selection. Lastly, the accuracy of the estimator also depends on the approximation of its integral $\hat{F}$. We already explained in the first example why we decided to use the composite Simpson's rule for

computing $\int \hat{f}(y)dy$ numerically, but it could be worth trying more complex and accurate quadrature formulas when performances are not a priority.

Regardless of the particular implementation of the KDE, the approach of pairing it with a robust NLP solver like WORHP or IPOPT has been proven to be able to handle a good variety of chance-constrained optimization problems in the domain of aerospace engineering. We are confident that the content of this chapter will reveal itself useful for the development of future research in this field.

Lastly, we remark that a natural link between the two parts of this thesis is the opportunity to combine the two approaches (i.e. Policy Iteration and Kernel Density Estimation) to study the implementation of an efficient numerical method for solving chance-constrained hybrid control problems.

Bibliography

[1] A. Agrachev, Y. Baryshnikov, and A. Sarychev. Ensemble controllability by lie algebraic methods. *ESAIM: Control, Optimisation and Calculus of Variations*, Publication ahead of print, 2016.

[2] L. Andrieu, G. Cohen, and F. J. Vásquez-Abad. Stochastic programming with probability constraints. *arXiv*, 0708.0281, 2007.

[3] M. Bardi and I. Capuzzo Dolcetta. *Optimal Control and Viscosity Solutions of Hamilton-Jacobi-Bellman Equations*. Birkhäuser Mathematics. Birkhauser, 1997.

[4] G. Barles, S. Dharmatti, and M. Ramaswamy. Unbounded viscosity solutions of hybrid control systems. *ESAIM: Control, Optimisation and Calculus of Variations*, 16(1):176–193, 2010.

[5] G. Barles and E. R. Jakobsen. On the convergence rate of approximation schemes for hamilton-jacobi-bellman equations. *ESAIM: Mathematical Modelling and Numerical Analysis*, 36(1):33–54, 2002.

[6] G. Barles and P. E. Souganidis. Convergence of approximation schemes for fully nonlinear second order equations. *Asymptotic Analysis*, 4(3):271–283, 1991.

[7] V. Baudoi. *Optimisation robuste multiobjectifs par modèles de substitution*. PhD thesis, 'Institut Supérieur de l'Aéronautique et de l'Espace, 2012.

[8] R. E. Bellman. *Dynamic Programming*. Princeton University Press, 1957.

[9] A. Bensoussan and J. L. Menaldi. Hybrid control and dynamic programming. *Dynamics of Continuous, Discrete and Impulsive Systems - Series B: Application and Algorithm*, 3(4):395–442, 1997.

[10] D. Berstimas. The price of robustness. *Operations Research*, 52(1):35–53, 2004.

[11] O. Bokanowski, S. Maroso, and H. Zidani. Some convergence results for howard's algorithm. *SIAM Journal on Numerical Analysis*, 47(4):3001–3026, 2009.

[12] J. F. Bonnans, S. Maroso, and H. Zidani. Error estimates for a stochastic impulse control problem. *Applied Mathematics and Optimization*, 55(3):327–357, 2007.

[13] J. F. Bonnans, P. Martinon, and E. Trélat. Singular arcs in the generalized goddard's problem. *Journal of Optimization Theory and Applications*, 139(2):439–461, 2008.

[14] J. F. Bonnans and A. Shapiro. Optimization problems with perturbations: A guided tour. *SIAM Review*, 40(2):228–264, 1998.

[15] 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(1):31–45, 1998.

[16] F. Camilli and M. Falcone. An approximation scheme for the optimal control of diffusion processes. *ESAIM: Mathematical Modelling and Numerical Analysis*, 29(1):97–122, 1995.

[17] I. Capuzzo Dolcetta and H. Ishii. Approximate solutions of the bellman equation of deterministic control theory. *Applied Mathematics and Optimization*, 11(1):161–181, 1984.

[18] P. Carpentier, J.-P. Chancelier, and G. Cohen. Optimal control under probability constraint. SADCO Kick off, 2011.

[19] A. Chacon and A. Vladimirsky. Fast two-scale methods for eikonal equations. *SIAM Journal on Scientific Computing*, 34(2):A547–A578, 2012.

[20] J. Chai and R. G. Sanfelice. Hybrid feedback control methods for robust and global power conversion. *IFAC-PapersOnLine*, 48(27):298–303, 2015.

[21] 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.

[22] J.-C. Culioli and G. Cohen. Optimisation stochastique sous contraintes en espérance. *Comptes Rendus de l'Académie des Sciences - Série 1*, 320(6):753–758, 1995.

[23] J. M. Danskin. *The Theory of Max-Min and its Application to Weapons Allocation Problems*, volume 5 of *Economics and Operations Research*. Springer, 1967.

[24] V. F. Dem'yanov and V. N. Malozemov. *Introduction to Minimax*. Wiley, 1974.

[25] D. Dentcheva. Optimization models with probabilistic constraints. In F. Dabbene G. Calafiore, editor, *Probabilistic and Randomized Methods for Design under Uncertainty*. Springer, 2003.

[26] S. Dharmatti and M. Ramaswamy. Hybrid control systems and viscosity solutions. *SIAM Journal on Contol and Optimization*, 44(1):1259–1288, 2005.

[27] M. Falcone and R. Ferretti. *Semi-Lagrangian approximation schemes for linear and Hamilton-Jacobi equations*, volume 133 of *Other Titles in Applied Mathematics*. SIAM, 2014.

[28] R. Ferretti and H. Zidani. Monotone numerical schemes and feedback construction for hybrid control systems. *Journal of Optimization Theory and Applications*, 165(2):507–531, 2014.

[29] A. Geletu, Klöppel M., H. Zhang, and P. Li. Advances and applications of chance-constrained approaches to systems optimisation under uncertainty. *International Journal of Systems Science*, 44(7):1209–1232, 2012.

[30] G. P. Gerdan and R. E. Deakin. Transforming cartesian coordinates x, y, z to geographical coordinates ϕ, λ, h. *The Australian Surveyor*, 44(1):55–63, 1999.

[31] R. H. Goddard. A method of reaching extreme altitudes. *Smithsonian Miscellaneous Collections*, 71(2):2–69, 1921.

[32] E. Hairer, S. P. Nørsett, and G. Wanner. *Solving Ordinary Differential Equations I: Nonstiff Problems*, volume 1 of *Springer Series in Computational Mathematics*. Springer, 2-nd edition, 1993.

[33] R. Henrion and W. Römisch. Hölder and lipschitz stability of solution sets in programs with probabilistic constraints. *Mathematical Programming*, 100(3):589–611, 2004.

[34] R. Henrion and C. Strugarek. Convexity of chance constraints with independent random variables. *Computational Optimization and Applications*, 41(2):263–276, 2008.

[35] R. A. Howard. *Dynamic Programming and Markov Processes*. MIT Press, 1960.

[36] K. Ishii. Viscosity solutions of nonlinearsecond order elliptic pdes associated with impulse control problems ii. *Funkcialaj Ekvacioj*, 38(2):297–328, 1995.

[37] E. R. Jakobsen and K. H. Karlsen. Continuous dependence estimates for viscosity solutions of fully nonlinear degenerate elliptic equations. *Electronic Journal of Differential Equations*, 2002(39):1–10, 2002.

[38] N. V. Krylov. On the rate of convergence of fnite difference approximation for bellman's equation. *St. Petersburg Mathematical Journal*, 9(3):639–650, 1998.

[39] N. V. Krylov. On the rate of convergence of fnite difference approximation for bellman's equation with variable coefficients. *Probability Theory and Related Fields*, 117(1):1–16, 2000.

[40] H. Kushner and P.G. Dupuis. *Numerical methods for stochastic control problems in continuous time*, volume 24 of *Stochastic Modelling and Applied Probability*. Springer, 2001.

[41] P. L'Ecuyer and G. Yin. Budget-dependent convergence rate of stochastic approximation. *SIAM Journal on Optimization*, 8(1):217–247, 1998.

[42] U. Ledzewicz and H. Schättler. Optimal bang-bang controls for a two-compartment model in cancer chemotherapy. *Journal of Optimization Theory and Applications*, 114(3):609–637, 2002.

[43] E. S. Levitin. On differential properties of the optimal value of parametric problems of mathematical programming. *Doklady Akademii Nauk SSSR*, 224:1354–1358, 1975.

[44] E. S. Levitin. Differentiability with respect to a parameter of the optimal value in parametric problems of mathematical programming. *Kibernetika*, 1:44–59, 1976.

[45] C. Louembet, D. Arzelier, and G. Deaconu. Robust rendezvous planning under maneuver execution errors. *Journal of Guidance, Control, and Dynamics*, 38(1):76–93, 2015.

[46] Y. Mansour and S. Singh. On the complexity of policy iteration. *Proceedings of the Fifteenth Conference on Uncertainty in Artificial Intelligence*, pages 401–408, 1999.

[47] K. Marti. Differentiation formulas for probability functions: The transformation method. *Mathematical Programming*, 75(2):201–220, 1996.

[48] N. Metropolis. The beginning of the monte carlo method. *Los Alamos Science*, Special Issue:125–130, 1987.

[49] É. A. Nadaraya. On non-parametric estimates of density functions and regression curves. *Theory of Probability & Its Applications*, 10(1):186–190, 1965.

[50] A. Nemirovski and A. Shapiro. Convex approximations of chance constrained programs. *SIAM Journal on Optimization*, 17(4):969–996, 2006.

[51] A. Prékopa. On probabilistic constrained programming. *Proceedings of the Princeton Symposium on Mathematical Programming*, pages 113–138, 1970.

[52] A. Prékopa. Contributions to the theory of stochastic programming. *Mathematical Programming*, 4(1):202–221, 1973.

[53] A. Prékopa. *Stochastic programming*. Kluwer Academic Publishers, 1995.

[54] A. Prékopa. Probabilistic programming. In A. Shapiro A. Ruszczuǹski, editor, *Stochastic programming*, volume 10. Elsevier, 2003.

[55] M. L. Puterman and S. L. Brumelle. On the convergence of policy iteration in stationary dynamic programming. *Mathematics of Operations Research*, 4(1):60–69, 1979.

[56] M. L. Puterman and M. C. Shin. Modified policy iteration algorithms for discounted markov decision problems. *Management Science*, 24(11):1127–1137, 1978.

[57] E. Raik. Qualitative research into the stochastic nonlinear programming problems. *Eesti NSV Teaduste Akademia Toimetised*, 20:8–14, 1971.

[58] S. Ross. *A First Course in Probability*. Pearson, 9-th edition, 2014.

[59] M. S. Santos and J. Rust. Convergence properties of policy iteration. *SIAM Journal on Control and Optimization*, 42(6):2094–2115, 2004.

[60] A. Sassi. Tecniche di programmazione dinamica nell'ottimizzazione di sistemi di controllo ibridi. Master's thesis, Università degli Studi Roma Tre, 2013.

[61] R. Serra. *Opérations de proximité en orbite : évaluation du risque de collision et calcul de manoeuvres optimales pour l'évitement et le rendez-vous*. PhD thesis, INSA Toulouse, 2015.

[62] R. Serra, D. Arzelier, M. Joldes, and A. Rondepierre. Probabilistic collision avoidance for long-term space encounters via risk selection. In J. Bordeneuve-Guibé, A. Drouin, and C. Roos, editors, *Advances in Aerospace Guidance, Navigation and Control*. Springer, 2015.

[63] J. A. Sethian and A. Vladimirsky. Fast methods for the eikonal and related hamilton-jacobi equations on unstructured meshes. *Proceedings of the National Academy of Sciences of United States of America*, 97(11):5699–5703, 2000.

[64] J. A. Sethian and A. Vladimirsky. Ordered upwind methods for hybrid control. In C. J. Tomlin and M. R. Greenstreet, editors, *Hybrid Systems: Computation and Control*, volume 2289. Springer, 2002.

[65] S. J. Sheather. Density estimation. *Statistical Science*, 19(4):588–597, 2004.

[66] B. W. Silverman. *Density Estimation for Statistics and Data Analysis*. Chapman & Hall, 1986.

[67] S. Uryasev. Derivatives of probability functions and some applications. *Annals of Operations Research*, 56(1):287–311, 1995.

[68] W. van Ackooij and R. Henrion. Gradient formulae for nonlinear probabilistic constraints with gaussian and gaussian-like distributions. *SIAM Journal on Optimization*, 24(4):1864–1889, 2014.

[69] A. Wächter and L. T. Biegler. On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. *Mathematical Programming*, 106(1):25–57, 2006.

[70] A. Wald. Contributions to the theory of statistical estimation and testing hypotheses. *The Annals of Mathematics*, 10(4):299–326, 1939.

[71] D. M. Young and R. T. Gregory. *A survey of numerical mathematics*. Addison-Wesley, 1972.

[72] H. Zhang and M.R. James. Optimal control of hybrid systems and a system of quasi-variational inequalities. *SIAM Journal on Control and Optimization*, 45(2):722–761, 2006.

université PARIS-SACLAY | École doctorale de mathématiques Hadamard (EDMH)

Titre : Méthodes numériques pour des problèmes de contrôle hybride et optimisation avec contraintes en probabilité
Mots clés : Contrôle optimal, Systèmes hybrides, Schémas numériques, Optimisation stochastique, Contrainte en probabilité, Lanceurs spatiaux
Résumé : Cette thèse est dediée à l'analyse numérique de méthodes numériques dans le domaine du contrôle optimal, et est composée de deux parties. La première partie est consacrée à des nouveaux résultats concernant des méthodes numériques pour le contrôle optimal de systèmes hybrides, qui peuvent être contrôlés simultanément par des fonctions mesurables et des sauts discontinus dans la variable d'état. La deuxième partie est dédiée à l'étude d'une application spécifique sur l'optimisation de trajectoires pour des lanceurs spatiaux avec contraintes en probabilité. Ici, on utilise des méthodes d'optimisation nonlineaires couplées avec des techniques de statistique non paramétrique. Le problème traité dans cette partie appartient à la famille des problèmes d'optimisation stochastique et il comporte la minimisation d'une fonction de coût en présence d'une contrainte qui doit être satisfaite dans les limites d'un seuil de probabilité souhaité.

Title : Numerical methods for hybrid control and chance-constrained optimization problems
Keywords : Optimal control, Hybrid systems, Numerical schemes, Stochastic optimization, Chance constraint, Space launchers
Abstract : This thesis is devoted to the analysis of numerical methods in the field of optimal control, and it is composed of two parts. The first part is dedicated to new results on the subject of numerical methods for the optimal control of hybrid systems, controlled by measurable functions and discontinuous jumps in the state variable simultaneously. The second part focuses on a particular application of trajectory optimization problems for space launchers. Here we use some nonlinear optimization methods combined with non-parametric statistics techniques. This kind of problems belongs to the family of stochastic optimization problems and it features the minimization of a cost function in the presence of a constraint which needs to be satisfied within a desired probability threshold.

Université Paris-Saclay
Espace Technologique / Immeuble Discovery
Route de l'Orme aux Merisiers RD 128 / 91190 Saint-Aubin, France