HAL
open science

Solving chance constrained optimal control problems in aerospace via Kernel Density Estimation

Jean-Baptiste Caillau, Max Cerf, Achille Sassi, Emmanuel Trélat, Hasnaa Zidani

To cite this version:

Jean-Baptiste Caillau, Max Cerf, Achille Sassi, Emmanuel Trélat, Hasnaa Zidani. Solving chance constrained optimal control problems in aerospace via Kernel Density Estimation. Optimal Control Applications and Methods, 2018, 39 (5), pp.1833-1858. 10.1002/oca.2445 . hal-01507063

HAL Id: hal-01507063

https://inria.hal.science/hal-01507063

Submitted on 12 Apr 2017

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

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

Solving chance constrained optimal control problems in aerospace via Kernel Density Estimation*

J.-B. Caillau[1], M. Cerf[2], A. Sassi[3], E. Trélat[4] and H. Zidani[3]

[1]Université de Bourgogne Franche-Comté
[2]Airbus Safran Launchers
[3]ENSTA ParisTech
[4]Université Pierre et Marie Curie

April 12, 2017

Abstract

The goal of this paper is to show how non-parametric statistics can be used to solve some chance constrained optimization and optimal control problems. We use the Kernel Density Estimation method to approximate the probability density function of a random variable with unknown distribution, from a relatively small sample. We then show how this technique can be applied and implemented for a class of problems including the Goddard problem and the trajectory optimization of an Ariane 5-like launcher.

Keywords: Kernel Density Estimation, chance constrained optimization, stochastic optimization, Optimal control, Aerospace engineering

1 Introduction

This paper is dedicated to a numerical approach for solving chance constrained optimal control problems, using the Kernel Density Estimation technique. One of the earliest and most famous examples of optimal control problems in aerospace dates back to the beginning of the twentieth century. In 1921, American physicist Robert HutchingsGoddard published a paper [13] 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 will give a simplified model. Consider the vertical ascent of a rocket in one dimension. The function $r(t)$ represents the rocket's altitude, $v(t)$ its speed, and

*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. For the third and fifth author, also by iCODE Institute project funded by the IDEX Paris-Saclay, ANR-11-IDEX-0003-02.

$m(t)$ its mass. We introduce the variable $u(t) \in [0,1]$ which defines the rate 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 $m(t)g$ with a fuel consumption rate $\frac{T}{v_\mathrm{e}}u(t)$ where v_e is the fuel speed. The maximum thrust T, the fuel speed v_e, the initial mass m_0 and the final time t_f are fixed. The controlled system describing the launcher's dynamics is

$$\begin{cases} \dot{r}(t) = v(t) & t \in [0, t_\mathrm{f}] \\ \dot{v}(t) = \frac{T}{m(t)}u(t) - g & t \in [0, t_\mathrm{f}] \\ \dot{m}(t) = -\frac{T}{v_\mathrm{e}}u(t) & t \in [0, t_\mathrm{f}] \\ \big(r(0), v(0), m(0)\big) = (0, 0, m_0) \end{cases} \tag{1.1}$$

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 ensuring that it reaches at least a given altitude $\bar{r}_\mathrm{f}$ at time t_f. Formally, this translates to solving

$$\begin{cases} \max\limits_{u \in \mathcal{U}} m_\mathrm{f}(u) \\ r_\mathrm{f}(u) \geq \bar{r}_\mathrm{f} \\ m_\mathrm{f}(u),\ r_\mathrm{f}(u) \text{ final mass and altitude associated to } u \text{ by (1.1).} \end{cases} \tag{1.2}$$

For a more general theoretical study of this kind of problems, we refer to [38]. Robust methods are aimed at achieving consistent performance and/or stability in the presence of bounded modelling errors. Drawing a parallel with the example, let us suppose that the thrust T is estimated with some margin errors, and assume that we want to maximize the final payload in presence of these uncertainties on T. One possible approach consists in the use of worst-case analysis to treat uncertainties in order to obtain what is called a "robust" solution. Using a variation of Wald's maximin model [39], one can consider the so-called *robust optimal control problem*:

$$\begin{cases} \max\limits_{u \in \mathcal{U}} \min\limits_{T \in [T_-, T_+]} m_\mathrm{f}(u, T) \\ r_\mathrm{f}(u, T) \geq \bar{r}_\mathrm{f} \qquad \forall T \in [T_-, T_+] \\ m_\mathrm{f}(u, T),\ r_\mathrm{f}(u, T) \text{ associated to } (u, T) \text{ by (1.1)} \end{cases} \tag{1.3}$$

A solution to this problem would be 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 for the final altitude for any $T \in [T_-, T_+]$. As pointed out in [3], robust optimization requires a trade-off: The price to obtain 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.

Another approach 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.

More precisely, in the case of our example, we assume that T is 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. We introduce the parameter $p \in [0,1]$ and consider the following problem

$$\begin{cases} \max\limits_{u \in \mathcal{U}} \mathbb{E}\big[m_{\mathrm{f}}(u,T)\big] \\ \mathbb{P}\big[r_{\mathrm{f}}(u,T) \geq \bar{r}_{\mathrm{f}}\big] \geq p \\ m_{\mathrm{f}}(u,T),\ r_{\mathrm{f}}(u,T) \text{ final mass and altitude associated to } (u,T) \text{ by (1.1)} \end{cases}$$

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. Optimal control problems with chance constraints are often considered if there is a need for minimizing a cost associated to the performance of a dynamical model, while taking into account uncertainties in the parameters defining it. (See also [30, 31] for other computations of probabilities in a dynamical setting related to aerospace engineering.) In this paper we study an efficient numerical solution to chance constrained optimal control problems in the form

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

where the functions J and G may depend on both a finite number of optimization variables x and on a control function u. The parameter $p \in (0,1)$ is a probability threshold and ξ is an m-dimensional random vector defined on some probability space that will be properly set later. We explore the application of the Kernel Density Estimation (KDE). This technique is 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: G being dependent on both x, u and ξ, it is not a trivial task to derive an analytical representation of its probability distribution, even if the distribution of ξ is known. The idea of applying non-parametric density estimation (and in particular KDE) to chance constrained optimization problems is not new (see for instance [29, 7] where this technique has been applied to an optimization problem in finite dimension). For the same type of problem, one can also mention [20] where the authors use a different technique, called *Scenario Approach*. The work in [41] features the use of KDE for solving problems in the simpler case where the optimization variables x are separated from the random variables ξ. To the best of our knowledge, however, KDE has not been used previously as a tool for solving optimal control problems, where both the cost and the constraints explicitly depend on a control function u.

The goal of this paper is to show the relevance of KDE approach on some optimal control problems. This method can lead to very good results, hopefully inspiring new developments in the field of chance constrained optimal control. In the next Section 2 we present some existing results on the subject of chance constrained optimization. Section 3 gives an overview of the KDE technique and introduces the algorithm that we propose to use for solving chance constrained control problems. Section 4 consists of three numerical examples involving the

application of KDE to chance constrained optimization and optimal control problems.

2 Chance-constrained optimization problems: A short state-of-the-art

There exists a wide literature on the subject of chance constrained optimization, in particular in the case of problems involving only an array of decision variables x in $\mathcal{X} \subseteq \mathbb{R}^n$. As already mentioned, 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 constraints, and even in the case of a relaxation approach, it might be hard to make sure that the problem satisfies all the required controllability hypotheses. Suppose that 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 $\mathbb{R}^m$. Consider then the chance constrained problem:

$$\begin{cases} \min\limits_{x \in \mathcal{X}} \mathbb{E}\big[J(x,\xi)\big] \\ \mathbb{P}\big[G(x,\xi) \geq 0\big] \geq p \end{cases} \tag{2.1}$$

where $\mathcal{X} \subseteq \mathbb{R}^n$ is the admissible set for the decision variables x, $J : \mathbb{R}^n \to \mathbb{R}$ is an objective, $G : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}$ defines a constraint inequality, $p \in (0,1)$ is a probability threshold called *confidence level*. This kind of problem has been treated at least since the fifties [8]. A general theory, however, is due to Prékopa [23, 24], who also introduced the convexity theory based on logconcavity. Other contributions on logconcavity theory in stochastic programming can be found in [25, 26, 9]. 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 in [27, 24]. The continuity theorem proven in [27] that, if the functions $G_i(., y)$ are upper semi-continuous [respectively, continuous] then the function

$$\mathcal{G}(x) := \mathbb{P}\big[G(x,\xi) \geq 0\big]$$

is upper semi-continuous [respectively, continuous]. Additionally, the convexity theorem in [24] states that if G_i are quasi-concave functions and ξ has a log-concave probability distribution, then the function $\mathcal{G}$ is a log-concave function. Let us mention that the results presented in [27, 24] apply to the more general case of multiple joint constraints. In [37] 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 $\mathcal{G}$. The main feature of this result is that it opens the path to many numerical approaches based on descent algorithms. Moreover, obtaining the gradient of the chance constraint is a crucial step towards establishing first order necessary conditions for optimality (see also [36] and [17]). When the probability measure μ is not precisely known and it is replaced by an estimator $\nu \in \mathcal{P}(\mathbb{R}^m)$, the result proven in [15] gives the hypotheses under which it can be possible to obtain an estimate on the difference

between the solution of the original problem and the one where the measure μ has been replaced by ν.

Some numerical approaches. In the literature [6], chance constrained optimal control problems have also been treated with other techniques, such as the Scenario Approach previously mentioned. Another alternative technique for solving chance constrained problems is the Monte Carlo method. A Monte Carlo algorithm 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 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. This theorem can be applied to estimate probabilities 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 $i \in \{1, 2, \dots, n\}$

$$X_i = \begin{cases} 1 & \text{if } E \text{ realizes at the } i\text{-th try} \\ 0 & \text{else} \end{cases}$$

we can apply the law of large numbers 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 of this class of methods. They are usually easy to implement and can be easily 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 particular field: From physics to statistics, from biology to finance, as well as engineering and Artificial Intelligence. An important technique rising from the combination of the Monte Carlo method with the iterative gradient method is the Stochastic Stochastic Arrow-Hurwicz Algorithm (SAHA). This algorithm can be used to solve an optimization problem in the form

$$\begin{cases} \min_{x \in \mathcal{X}} \mathbb{E}\big[J(x, \xi)\big] \\ \mathbb{E}\big[H(x, \xi)\big] \leq \alpha \end{cases} \tag{2.2}$$

where the function H has to be regular enough to guarantee the convexity and the connectedness of the feasible subset defined by the constraint. (See [2] for more details.)

3 Kernel Density Estimation (KDE) for chance constrained optimization problems

The approach we present for solving numerically the chance constrained control problems is based on Kernel Density Estimation. This technique consists in approximating the probability density function (PDF) of a random variable with unknown distribution from a given sample.

3.1 An Overview of KDE techniques

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

$$\hat{f}_{n,h}(x) := \frac{1}{nh} \sum_{i=1}^{n} K\left(\frac{x - X_i}{h}\right) \tag{3.1}$$

where the function K is called *kernel* and the smoothing parameter h is called *bandwidth*. The earlier mentions of this method in its current form date back to the early 50s in the works of Rosemblatt [28] and Parzen [21], this is why it is also known as *Parzen-Rosenblatt window*. Silverman's book [34] represents the basic text on the subject, while [35] provides a detailed analysis on the various properties of this technique. This method has also been applied to many other fields like archaeology, banking, climatology, economics, genetics, hydrology and physiology (see [32] for more references). A fundamental consistency result was obtained by Nadaraya [19]. Variations of this theorem have been studied in [33, 10], while an earlier but less general version of this result can also be found in [21].

Theorem 3.1 (Nadaraya [19]). *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*

$$\mathbb{P}\Big[\lim_{n \to +\infty} \sup_x \big|\hat{f}_{n,h}(x) - f(x)\big| = 0\Big] = 1.$$

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

$$\int K(y)\,\mathrm{d}y = 1 \quad \text{and} \quad \int yK(y)\,\mathrm{d}y = 0 \quad \text{and} \quad \int y^2 K(y)\,\mathrm{d}y > 0.$$

Bandwidth and kernel selection. Let us first define the *bias* and *variance* the density estimator $\hat{f}_{n,h}(x)$:

$$\begin{aligned} \mathrm{Bias}\big[\hat{f}_{n,h}(x)\big] &:= \mathbb{E}\big[\hat{f}_{n,h}(x)\big] - f(x) \\ \mathrm{Var}\big[\hat{f}_{n,h}(x)\big] &:= \mathbb{E}\Big[\Big(f_{n,h}(x) - \mathbb{E}\big[\hat{f}_{n,h}(x)\big]\Big)^2\Big]. \end{aligned}$$

With these definitions we can use the Mean Integrated Squared Error (MISE) as a measure of the discrepancy between $\hat{f}$ and f:

$$\begin{aligned}\mathrm{MISE}\big[\hat{f}(x)\big] &:= \mathbb{E}\left[\int \big(\hat{f}_{n,h}(x) - f(x)\big)^2 \,\mathrm{d}x\right] = \\ &= \int \mathrm{Bias}^2\big[\hat{f}_{n,h}(y)\big]\,\mathrm{d}y + \int \mathrm{Var}\big[\hat{f}_{n,h}(y)\big]dy.\end{aligned}$$

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

$$\begin{aligned}\mathrm{Bias}\big[\hat{f}_{n,h}(x)\big] &= \frac{h^2}{2}\int y^2 K(y)\,\mathrm{d}y f''(x) + o(h^2) \\ \mathrm{Var}\big[\hat{f}_{n,h}(x)\big] &= \frac{1}{nh}\int K^2(y)\,\mathrm{d}y f(x) + o\left(\frac{1}{nh}\right).\end{aligned}$$

Under integrability assumptions on f (see [32]), we can define the main term in the Taylor expansion of the MISE as the Asymptotic Mean Integrated Squared Error (AMISE):

$$\mathrm{AMISE}\big[\hat{f}_{n,h}(x)\big] := \frac{1}{nh}\int K^2(y)\,\mathrm{d}y + \frac{h^4}{4}\left(\int y^2 K(y)\,\mathrm{d}y\right)^2 \int f''^2(y)\,\mathrm{d}y \quad (3.2)$$

which leads to the following choice for the bandwidth h minimizing (3.2)

$$h_{\mathrm{AMISE}} := \sqrt[5]{\frac{\int K^2(y)\,\mathrm{d}y}{n\left(\int y^2 K(y)\,\mathrm{d}y\right)^2 \int f''^2(y)\,\mathrm{d}y}}. \quad (3.3)$$

If we substitute the optimal bandwidth given by (3.3) in (3.2), we obtain

$$\mathrm{AMISE}\big[\hat{f}_{n,h_{\mathrm{AMISE}}}(x)\big] = \frac{5}{4}\sqrt[5]{\frac{\left(\int K^2(y)\,\mathrm{d}y\right)^4\left(\int y^2 K(y)\,\mathrm{d}y\right)^2 \int f''^2(y)\,\mathrm{d}y}{n^4}}. \quad (3.4)$$

Showing that the AMISE will tend to zero at a rate $n^{-\frac{4}{5}}$. Unfortunately though, the presence of the unknown factor $\int f''^2(y)\,\mathrm{d}y$ in (3.3) makes the expression of h_{AMISE} almost useless. For this reason, it might be more viable to approximate also the derivatives of f [33, 4] or use one of the many practical ways [32, 34] for choosing the bandwidth using only information from the sample. Other results on the rate of convergence of the KDE have been proved in [12, 22, 1, 18].

On a more practical side, 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

$$S := \sqrt{\frac{1}{n}\sum_{1=1}^{n}\left(X_i - \frac{\sum_{1=1}^{n} X_i}{n}\right)}$$

the SNR bandwidth is then defined as

$$h_{\mathrm{SNR}} := 1.06\frac{S}{\sqrt[5]{n}}. \quad (3.5)$$

Even though there is no general rule for obtaining an explicit value of h leading to the best approximation of f, it is 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. As for the choice of the kernel, we want to show that when using the AMISE expression for the approximation error, there is little room for improvement. In order to do this we need to define the *efficiency* associated to a kernel. We first define the function

$$C(K) := \sqrt[5]{\left(\int y^2 K(y)\,\mathrm{d}y\right)^2 \left(\int K^2(y)\,\mathrm{d}y\right)^4}.$$

Substituting it in (3.4), we have that minimizing (3.4) with respect to K is equivalent to minimizing

$$\frac{5}{4} C(K) \sqrt[5]{\frac{\int f''^2(y)\,\mathrm{d}y}{n^4}}.$$

This means that we should consider kernels with small values of $C(K)$. If we focus on kernels that are themselves probability density functions (which are the only ones ensuring that the estimate is everywhere non-negative), we have $\int K(y)\,\mathrm{d}y = 1$. Moreover, we can also assume $\int y^2 K(y)\,\mathrm{d}y = 1$: The fact that K is a density function guarantees that $\int y^2 K(y)\,\mathrm{d}y$ is finite, thus allowing us to choose its normalized version in case $\int y^2 K(y)\,\mathrm{d}y \neq 1$. Since our kernel satisfies

$$\int K(y)\,\mathrm{d}y = \int y^2 K(y)\,\mathrm{d}y = 1 \tag{3.6}$$

minimizing $C(K)$ reduces to minimizing $\int K^2(y)\,\mathrm{d}y$, and in [16] it has been proven that the kernel

$$K_{\mathrm{e}}(y) := \begin{cases} \frac{3}{4\sqrt{5}}\left(1 - \frac{1}{5}y\right) & |y| \leq \sqrt{5} \\ 0 & \text{else} \end{cases} \tag{3.7}$$

achieves the minimal value of $C(K)$ under the constraints (3.6). The efficiency of any kernel K satisfying (3.6) is then defined as

$$\mathrm{eff}(K) := \left(\frac{C(K_{\mathrm{e}})}{C(K)}\right)^{\frac{5}{4}} = \frac{3}{5\int K^2(y)\,\mathrm{d}y \sqrt{5\int y^2 K(y)\,\mathrm{d}y}}$$

where K_e is the Epanechniov kernel, defined in (3.7). Table 1 reports the efficiency of some of the most used kernels. Note that even the rectangular kernel (arguably the most naive choice of K) achieves an efficiency of ≈ 0.93. This leads us to the conclusion that, when measuring the error by means of (3.2), the choice of the kernel is not as important as the choice of the bandwidth h.

3.2 Application of KDE approach to chance constrained optimization

Consider the following problem

$$\begin{cases} \min\limits_{(x,u)\in\mathcal{X}\times\mathcal{U}} J(x,u) \\ \mathbb{P}\big[G(x,u,\xi) \geq 0\big] \geq p. \end{cases} \tag{3.8}$$

Kernel	$K(y)$	eff(K)
Epanechniov	$\begin{cases} \frac{3}{4\sqrt{5}}\left(1-\frac{1}{5}y\right) & \lvert y\rvert \leq \sqrt{5} \\ 0 & \text{else} \end{cases}$	≈ 0.9939
Biweight	$\begin{cases} \frac{15}{16}(1-y^2)^2 & \lvert y\rvert \leq 1 \\ 0 & \text{else} \end{cases}$	≈ 0.9859
Triangular	$\begin{cases} 1-\lvert y\rvert & \lvert y\rvert \leq 1 \\ 0 & \text{else} \end{cases}$	≈ 0.9295
Gaussian	$\frac{1}{\sqrt{2\pi}}e^{-\frac{y^2}{2}}$	≈ 0.9512
Rectangular	$\begin{cases} \frac{1}{2} & \lvert y\rvert \leq 1 \\ 0 & \text{else} \end{cases}$	≈ 0.9295

Table 1: Efficiency of some kernels

where $\mathcal{X} \subseteq \mathbb{R}^d$ and $\mathcal{U}$ are respectively the admissible sets for the decision variables x and the control u, $J : \mathbb{R}^d \times \mathcal{U} \to \mathbb{R}$ and $G : \mathbb{R}^d \times \mathcal{U} \times \mathbb{R}^m \to \mathbb{R}$ are respectively the cost and constraint functions. By using KDE, we are able to produce an approximation of the PDF defining the chance constraint, thus allowing us to replace the probability with the integral of the estimated PDF and solve the stochastic optimization problem as a deterministic one. For given x in $\mathcal{X}$ and u in $\mathcal{U}$, let $f_{x,u}$ and $\hat{f}_{x,u}$ denote respectively the PDF of $G(x, u, \xi)$ and its approximation. One has

$$\mathbb{P}\big[G(x,u,\xi) \geq 0\big] = 1 - \mathbb{P}\big[G(x,u,\xi) < 0\big] = 1 - \int_{-\infty}^{0} f_{x,u}(z)\,\mathrm{d}z.$$

We then build the estimator $\hat{f}_{x,u}$ of $f_{x,u}$ via KDE. By defining $\hat{F}_{x,u}(y) := \int_{-\infty}^{y} \hat{f}_{x,u}(z)\,\mathrm{d}z$ we can write an approximation of our chance constraint in the form

$$\begin{cases} \min\limits_{(x,u)\in\mathcal{X}\times\mathcal{U}} J(x,u) \\ \hat{F}_{x,u}(0) \leq 1-p. \end{cases} \tag{3.9}$$

Let (x^*, u^*) and $(\hat{x}^*, \hat{u}^*)$ be respectively the solutions of problems (3.8) and (3.9). Even in absence of an explicit estimate for the error between x^* and $\hat{x}^*$ by means of the error between $\hat{f}_{\hat{x}^*,\hat{u}^*}$ and f_{x^*,u^*}, we can always rely on the law of large numbers for the validation of our results a posteriori. The detailed procedure for implementing the approximation of the distribution $F_{x,u}$ in practice following the steps below.

1- Draw the sample. Take a sample of size n from the random vector ξ: $\{\xi_1, \xi_2, \ldots, \xi_n\}$, where each ξ_i is an m-dimensional array. Note that this operation has to be done only once at the beginning of the optimization procedure, since the realizations of ξ are only dependent on its distribution and not on the decision variables and control x and u.

2- Define the constraint function. Now we need to define the value of $\hat{F}_{x,u}$ for any value of x and u.

1. Given $(x, u) \in \mathcal{X} \times \mathcal{U}$, for each element ξ_i of the sample, compute the constraint function $G(x, u, \xi_i)$.

2. Estimate via KDE the distribution of the new sample

$$\{G(x,u,\xi_1), G(x,u,\xi_2), \dots, G(x,u,\xi_n)\}.$$

Choose a kernel K, a bandwidth h and define

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

3. Compute $\hat{F}_{x,u}(s) := \int_{-\infty}^{s} \hat{f}_{x,u}(y)\,\mathrm{d}y$.

Note that this step has to be repeated every time we need to evaluate the function $\hat{F}_{x,u}$ (and thus G) for different values of (x,u).

3- Solve the approximated problem. Now that the approximation $\hat{F}_{x,u}$ of $F_{x,u}$ has been defined, we can solve problem (3.9) as a regular deterministic optimization problem.

4- Validate the solution We use Borel's law of large numbers to check the quality of solutions. For each n, we will call (x^*, u^*) the optimal solution found at the previous step and then draw a large random sample of size N_a from ξ. Let N_s be the number of times that the event $G(x^*, u^*, \xi) \geq 0$ occurs. Simplifying, Borel's law of large numbers states that

$$\lim_{N_\mathrm{s}\to+\infty} \frac{N_\mathrm{s}}{N_\mathrm{a}} = \mathbb{P}[G(x^*, u^*, \xi) \geq 0].$$

4 Numerical tests

This section contains some numerical applications of the KDE. The first example involves a standard chance constrained optimization problem with a finite number of optimization variables. In the second example though, we treat an optimal control problem in which both the cost and the constraint functions also depend on a measurable control. Finally, the third example considers a more realistic and complex control problem of multi-stage launcher. For the examples in this section, 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

$$\int_a^b f(y)\,\mathrm{d}y \approx \frac{1}{3}\frac{b-a}{N}\left(f(a) + 2\sum_{i=1}^{\frac{N}{2}-1} f(y_{2i}) + 4\sum_{i=1}^{\frac{N}{2}} f(y_{2i-1}) + f(b)\right)$$

where $y_i := a + \frac{b-a}{N} i \quad \forall i \in \{0, 1, \dots, N-1\}$. We decided to use Simpson's rule for this method because it provides a good balance between ease of code implementation and precision, since the error of this quadrature formula is bounded by $\left(\frac{b-a}{N}\right)^4 (b-a) \max_{y\in[a,b]} \left|f^{(4)}(y)\right|$. Details and results on this formula can be found in [40].

Nonlinear optimization solvers used in the numerical simulations. The results in this section have been obtained by using Fortran 90 to write the code interface and WORHP as solver. This solver is designed to handle finite dimensional nonlinear optimization problem in the form

$$\begin{cases} \min\limits_{X \in \mathbb{R}^N} F(X) \\ X_{\mathrm{L}} \leq X \leq X_{\mathrm{U}} \\ G_{\mathrm{L}} \leq G(X) \leq G_{\mathrm{U}} \end{cases} \tag{4.1}$$

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 (4.1), while the derivatives of F and G are optional since they can be 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, please refer to the *User's Guide to WORHP* available at `www.worhp.de`. The tests in this section have been performed on a laptop equipped with an Intel i7-4558U CPU running at 2.8 GHz and 8 GB of RAM. Throughout the three examples the performances are mostly constant: the number of iterations required by WORHP to converge does not depend on n and it falls in the range of 5 to 10. On the other hand, the CPU time per iteration grows as n increases, varying from less than a tenth of a second when n is less than 100 to a maximum of 5 seconds for $n = 10000$.

4.1 Example 1: Chance-constrained fuel load optimization of a simple three-stage launcher

Model. We first consider a simple model for a three-stage launcher. We analyze the vertical ascent of a rocket consisting in three sections, each one has its own fuel load and engine. During the flight the vehicle will separate the empty structure of each stage as soon as the fuel load contained is completely exhausted. The ODE systems describing the dynamics of the i-th phase and the initial conditions are

$$\begin{cases} \dot{r}(t) = v(t) & t \in (t_{i-1}, t_i) \quad \text{(altitude)} \\ \dot{v}(t) = \frac{T_i}{m(t)} - g & t \in (t_{i-1}, t_i) \quad \text{(speed)} \\ \dot{m}(t) = -\frac{T_i}{v_{\mathrm{e}i}} & t \in (t_{i-1}, t_i) \quad \text{(mass)} \\ \big(r(0), v(0), m(0)\big) = (0, 0, m_0). \end{cases} \tag{4.2}$$

For each phase i we define the final time t_i, the engine thrust T_i and the fuel speed $v_{\mathrm{e}i}$. g is the gravitational acceleration of the Earth. The initial mass of the launcher is defined as the sum of the three stages fuel and structure, plus the payload:

$$m_0 = \sum_{i=1}^{3} (1 + k_i) m_{\mathrm{e}i} + m_{\mathrm{u}}$$

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 indexes and the fuel masses of the three stages. m_0 has to satisfy the inequality $m_0 < \frac{T}{g}$ to make sure that the launcher is not too heavy or, equivalently, that the engine is powerful enough to overcome the force of gravity. Moreover, the final time of each phase satisfies

$$t_1 = \frac{v_{e1} m_{\mathrm{e}1}}{T_1} \qquad t_2 = t_1 + \frac{v_{\mathrm{e}2} m_{\mathrm{e}2}}{T_2} \qquad t_3 = t_2 + \frac{v_{\mathrm{e}3}(m_{\mathrm{e}3} + m_\mathrm{u})}{T_3}.$$

In the definition of t_3, the payload m_u is summed to the fuel mass of the third stage. This allows 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 $\mathbf{v}_\mathrm{e} := (v_{\mathrm{e}1}, v_{\mathrm{e}2}, v_{\mathrm{e}3})$. In this simple model, for $0 \le t \le t_1$ 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 $t_1 < t \le t_2$

$$\begin{aligned}
r(t) &= \left(v_{\mathrm{e}1} t - \frac{v_{\mathrm{e}1}^2 m(0)}{T_1}\right) \ln\left(\frac{m(0)}{m(t_1)}\right) + \\
&+ \left(v_{\mathrm{e}2}(t - t_1) - \frac{v_{\mathrm{e}2}^2 (m(t_1) - k_1 m_{\mathrm{e}1})}{T_2}\right) \ln\left(\frac{m(t_1) - k_1 m_{\mathrm{e}1}}{m(t)}\right) + \\
&+ v_{\mathrm{e}1} t_1 + v_{\mathrm{e}2}(t - t_1) - \frac{g}{2} t^2 \\
v(t) &= v_{\mathrm{e}1} \ln\left(\frac{m(0)}{m(t_1)}\right) + v_{\mathrm{e}2} \ln\left(\frac{m(t_1) - k_1 m_{\mathrm{e}1}}{m(t)}\right) - gt \\
m(t) &= \sum_{i=2}^{3} (1 + k_i) m_{\mathrm{e}i} + \overline{m}_\mathrm{u} - \frac{T_2}{v_{\mathrm{e}2}} (t - t_1)
\end{aligned}$$

and lastly, if $t_2 < t \le t_3$

$$\begin{aligned}
r(t) &= \left(v_{\mathrm{e}1} t - \frac{v_{\mathrm{e}1}^2 m(0)}{T_1}\right) \ln\left(\frac{m(0)}{m(t_1)}\right) + v_{\mathrm{e}1} t_1 + \\
&+ \left(v_{\mathrm{e}2}(t - t_1) - \frac{v_{\mathrm{e}2}^2 (m(t_1) - k_1 m_{\mathrm{e}1})}{T_2}\right) \ln\left(\frac{m(t_1) - k_1 m_{\mathrm{e}1}}{m(t_2)}\right) + \\
&+ \left(v_{\mathrm{e}3}(t - t_2) - \frac{v_{\mathrm{e}3}^2 (m(t_2) - k_2 m_{\mathrm{e}2})}{T_3}\right) \ln\left(\frac{m(t_2) - k_2 m_{\mathrm{e}2}}{m(t)}\right) + \\
&+ v_{\mathrm{e}1} t_1 + v_{\mathrm{e}2}(t_2 - t_1) + v_{\mathrm{e}3}(t - t_2) - \frac{g}{2} t^2 \\
v(t) &= v_{\mathrm{e}1} \ln\left(\frac{m(0)}{m(t_1)}\right) + v_{\mathrm{e}2} \ln\left(\frac{m(t_1) - k_1 m_{\mathrm{e}1}}{m(t_2)}\right) + \\
&+ v_{\mathrm{e}3} \ln\left(\frac{m(t_2) - k_2 m_{\mathrm{e}2}}{m(t)}\right) - gt \\
m(t) &= (1 + k_3) m_{\mathrm{e}3} + \overline{m}_\mathrm{u} - \frac{T_3}{v_{\mathrm{e}3}} (t - t_2).
\end{aligned}$$

Before defining the stochastic optimization problem associated to this model, let us define a reference deterministic optimization problem.

$$\begin{cases} \min\limits_{\mathbf{m}_\mathrm{e} \in \mathbb{R}_+^3} \sum\limits_{i=1}^{3} (1+k_i) m_{\mathrm{e}i} + m_\mathrm{u} \\ M_\mathrm{u}(\mathbf{m}_\mathrm{e}) \geq m_\mathrm{u} \end{cases} \tag{4.3}$$

where the constraint function is defined as $M_\mathrm{u}(\mathbf{m}_\mathrm{e}) := m\big(t_3(\mathbf{m}_\mathrm{e})\big) - k_3 m_{\mathrm{e}3}$. For a given $\mathbf{m}_\mathrm{e}$, $t_3(\mathbf{m}_\mathrm{e})$ is the solution to the equation obtained by imposing a constraint on the apogee of the launcher's final position $r(t_3) + \frac{v^2(t_3)}{2g} = \omega_\mathrm{f}$. Table 2 sums up the choice of parameters. The optimal solution found is $m_{\mathrm{e}1}^* \approx 0.2153$,

Parameter	T_i	k_i	$v_{\mathrm{e}i}$	g	m_u	ω_f
Value	150	0.1	5	9.8	0.5	0.5

Table 2: Parameters for the deterministic optimization

$m_{\mathrm{e}2}^* \approx 0.1838$, $m_{\mathrm{e}3}^* \approx 0.0774$, with a corresponding optimal cost $\sum_{i=1}^{3}(1+k_i)m_{\mathrm{e}i}^* + \overline{m}_\mathrm{u} \approx 1.0241$. Figure 1 shows the corresponding optimal trajectory.

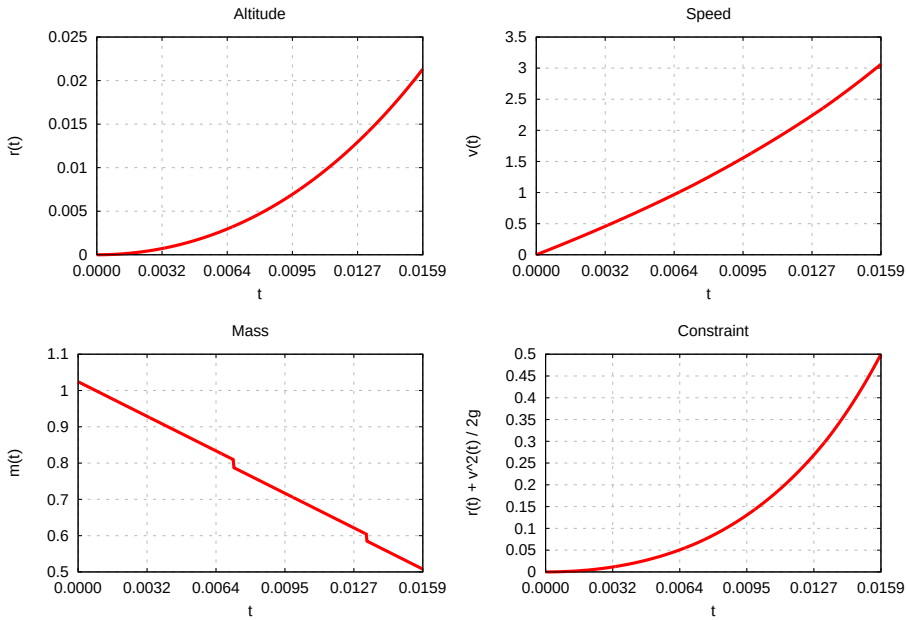


Figure 1: Plot of altitude, speed, mass and constraint for the three-stage launcher.

Problem statement. Let us now suppose that the parameters $\mathbf{T}$, $\mathbf{k}$ and $\mathbf{v}_\mathrm{e}$ are arrays of uniformly distributed random variables. For example, for each i in $\{1,2,3\}$, this implies $T_i \sim U(T_{i-}, T_{i+})$, where $T_{i-} := \overline{T}_i(1-\Delta T_i)$, $T_{i+} := \overline{T}_i(1+\Delta T_i)$ and $\overline{T}_i$ denotes the expected value. 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 same properties and definitions hold for $\mathbf{k}$

and $\mathbf{v}_{\mathrm{e}}$. If we want to write in form (3.8) the stochastic counterpart of problem (4.3), 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:

$$\mathbb{E}\left[\sum_{i=1}^{3}(1+k_i)m_{\mathrm{e}i}+m_{\mathrm{u}}\right]=\sum_{i=1}^{3}\big(1+\mathbb{E}\left[k_i\right]\big)m_{\mathrm{e}i}+m_{\mathrm{u}}.$$

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

$$\sum_{i=1}^{3}\big(1+\mathbb{E}\left[k_i\right]\big)m_{\mathrm{e}i}+m_{\mathrm{u}}=\sum_{i=1}^{3}(1+\overline{k}_i)m_{\mathrm{e}i}+m_{\mathrm{u}}.$$

This leads us to the stochastic optimization problem

$$\begin{cases}\min\limits_{\mathbf{m}_{\mathrm{e}}\in\mathbb{R}^3_+}\sum\limits_{i=1}^{3}(1+\overline{k}_i)m_{\mathrm{e}i}+m_{\mathrm{u}}\\ \mathbb{P}\big[M_{\mathrm{u}}(\mathbf{T},\mathbf{k},\mathbf{v}_{\mathrm{e}},\mathbf{m}_{\mathrm{e}})\geq m_{\mathrm{u}}\big]\geq p\end{cases}\tag{4.4}$$

with a total of nine uniform random variables (three random arrays of dimension three): $\mathbf{T}$, $\mathbf{k}$ and $\mathbf{v}_{\mathrm{e}}$. The function $M_{\mathrm{u}}(\mathbf{T},\mathbf{k},\mathbf{v}_{\mathrm{e}},\mathbf{m}_{\mathrm{e}})$ depends on the random arrays $\mathbf{T}$, $\mathbf{k}$ and $\mathbf{v}_{\mathrm{e}}$, and on the parameter $\mathbf{m}_{\mathrm{e}}$: $M_{\mathrm{u}}(\mathbf{T},\mathbf{k},\mathbf{v}_{\mathrm{e}},\mathbf{m}_{\mathrm{e}}) := m\big(t_3(\mathbf{T},\mathbf{k},\mathbf{v}_{\mathrm{e}},\mathbf{m}_{\mathrm{e}})\big)-k_3m_{\mathrm{e}3}$. Table 3 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	0.1	0.1	0.1	5	0.1

Table 3: Additional parameters for the stochastic optimization

Application of the method. We now have to reformulate the chance constraint showing its dependency on the CDF of the random variable M_{u}.

$$\begin{cases}\min\limits_{\mathbf{m}_{\mathrm{e}}\in\mathbb{R}^3_+}\sum\limits_{i=1}^{3}(1+\overline{k}_i)m_{\mathrm{e}i}+m_{\mathrm{u}}\\ F_{\mathbf{m}_{\mathrm{e}}}(m_{\mathrm{u}})\leq 1-p\end{cases}\tag{4.5}$$

where $F_{\mathbf{m}_{\mathrm{e}}}(m_{\mathrm{u}}) := \int_0^{m_{\mathrm{u}}} f_{\mathbf{m}_{\mathrm{e}}}(x)\,\mathrm{d}x$. 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

$$\begin{cases}\min\limits_{\mathbf{m}_{\mathrm{e}}\in\mathbb{R}^3_+}\sum\limits_{i=1}^{3}(1+\overline{k}_i)m_{\mathrm{e}i}+m_{\mathrm{u}}\\ \hat{F}_{\mathbf{m}_{\mathrm{e}}}(m_{\mathrm{u}})\leq 1-p\end{cases}\tag{4.6}$$

The procedure used for solving problem (4.6) is described in 3.2. We choose to use the SNR method (see 3.5) for computing the bandwidth combined with the Gaussian kernel.

Numerical results. Figure 2 shows 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 $N_{\mathrm{a}} = 10^5$. For this example, we decided to use samples of size up to 10000 because of the higher number of random variables the constraint function depends on. Figure 3 instead shows the the average value

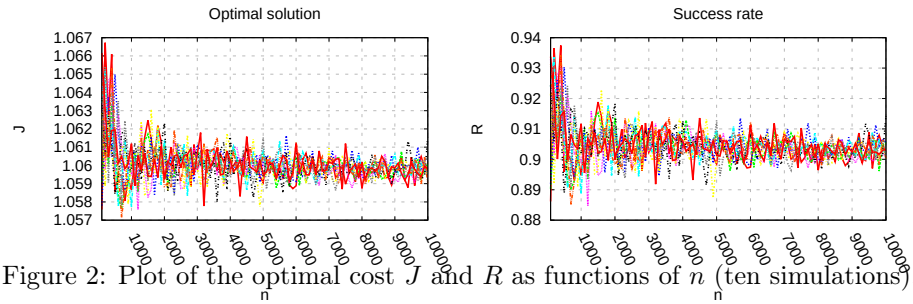


Figure 2: Plot of the optimal cost J and R as functions of n (ten simulations).

and variance of the ten sequences previously shown for each n. For instance, for

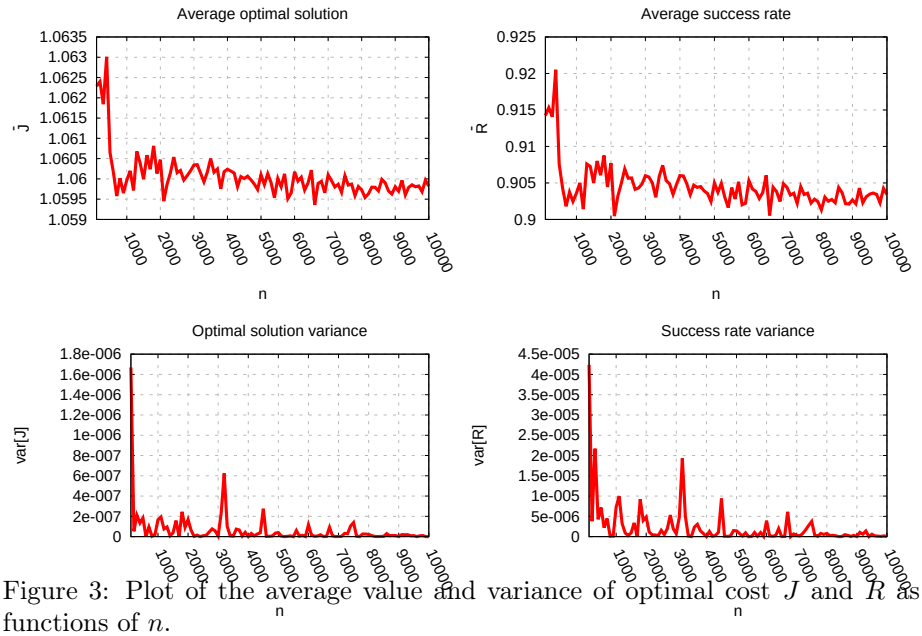


Figure 3: Plot of the average value and variance of optimal cost J and R as functions of n.

$n = 500$ the optimal solution is $m_{\mathrm{e}_1}^* \approx 0.2222$, $m_{\mathrm{e}_2}^* \approx 0.1835$, $m_{\mathrm{e}_3}^* \approx 0.1029$, with a corresponding optimal cost of $\sum_{i=1}^{3}(1+\overline{k}_i)m_{\mathrm{e}_i}^* + \overline{m}_{\mathrm{u}} \approx 1.0595$. This solution 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. Figure 4 shows the related plots. Table 4 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. Table 5 shows the comparison between the solution

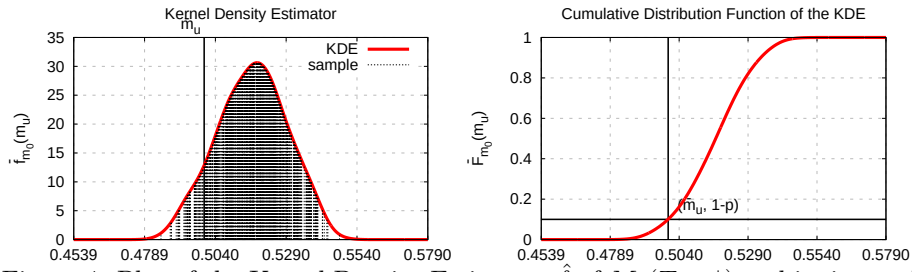


Figure 4: Plot of the Kernel Density Estimator $\hat{f}$ of $M_{\mathrm{u}}(T, \mathbf{m}_{\mathrm{e}}^*)$ and its integral $\hat{F}$.

Case	T_i	k_i	$v_{\mathrm{e}i}$	m_0^*
Random	$\sim U(T_{i-}, T_{i+})$	$\sim U(k_{i-}, k_{i+})$	$\sim U(v_{\mathrm{e}i-}, v_{\mathrm{e}i+})$	**1.0595**
Best	T_{i+}	k_{i-}	$v_{\mathrm{e}i+}$	**0.9497**
Worst	T_{i-}	k_{i+}	$v_{\mathrm{e}i-}$	**1.1246**

Table 4: Result comparison for extremal values of $\mathbf{T}$, $\mathbf{k}$ and $\mathbf{v}_{\mathrm{e}}$.

of the deterministic problem (4.3) and its stochastic counterpart (4.4) when p is close to 1 and ΔT_i, Δk_i and $\Delta v_{\mathrm{e}i}$ are close to 0. Unfortunately though, this method does not allow arbitrarily small values of ΔT_i, Δk_i or $\Delta v_{\mathrm{e}i}$. As reported in the table, when we don't provide enough variation to the sample, the success rate does not match the chosen probability. This is likely due to two issues related to the presence of the sample variance in (3.5), and therefore to ΔT_i, Δk_i or $\Delta v_{\mathrm{e}i}$. First, if they are too small, the Gaussian distributions summed in (3.1) tend to superimpose over the same points and do 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 $\Delta T_i = \Delta k_i = \Delta v_{\mathrm{e}i} = 0.1$ in Figure 4: Notice the space beneath the red graph on the left and right sides of the vertical sample lines. Secondly, because the bandwidth depends on the sample variance, the accuracy of the estimator might decrease if h is too small, as h appears as a denominator in (3.1). The results showed in Table 5 confirm that it is possible to increase the variance of the sample by increasing the number of random variables.

4.2 Example 2: Chance-constrained Goddard problem

We now apply the KDE technique to the Goddard problem. Formally, the structure of the model is the same as the example illustrated in the introduction: 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 (1.2) 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. In contrast with Example 1 which boiled down to a finite dimensional optimization problem, a solution now consists in an optimal control function $u^* : \mathbb{R}_+ \to [0, 1]$ such that, if we apply u^* irregardless of the value of T, the probability of the final altitude being greater

n	p	ΔT_i Δk_i $\Delta v_{\mathrm{e}i}$	h	$m_{\mathrm{e}1}$	$m_{\mathrm{e}2}$	$m_{\mathrm{e}3}$	m_0^*	R
Stochastic								
10^4	0.8	0.5	0.0117	0.218	0.191	0.202	**1.1714**	**0.7966**
		0.25	0.00508	0.226	0.185	0.124	**1.0887**	**0.8120**
		0.1	0.00191	0.237	0.158	0.103	**1.0477**	**0.8055**
		0.01	0.00019	0.214	0.166	0.098	**1.0259**	**0.8024**
		0.001	0.00002	0.209	0.167	0.101	**1.0239**	**0.8281**
	0.9	0.5	0.01472	0.186	0.281	0.271	**1.3115**	**0.9052**
		0.25	0.00517	0.227	0.205	0.133	**1.1215**	**0.8993**
		0.1	0.00192	0.218	0.184	0.106	**1.0596**	**0.9018**
		0.01	0.00018	0.215	0.166	0.098	**1.0271**	**0.9013**
		0.001	0.00002	0.214	0.166	0.097	**1.0241**	**0.9397**
	0.995	0.5	no convergence					
		0.25	0.00579	0.261	0.214	0.197	**1.2394**	**0.9960**
		0.1	0.00202	0.220	0.212	0.106	**1.0919**	**0.9951**
		0.01	0.00019	0.216	0.168	0.098	**1.0298**	**0.9959**
		0.001	0.00002	0.212	0.169	0.096	**1.0244**	**0.9996**
Deterministic								
—				0.215	0.184	0.077	1.0241	—

Table 5: Result comparison for different values of n, p and ΔT.

than r_{f} is greater than p.

Model. The original formulation of the Goddard problem can be found in [13]. We will consider a one-dimensional version of the one treated in [5].

The ODE system is

$$\begin{cases} \dot{r}(t) = v(t) & t \in [0, t_{\mathrm{f}}] \\ \dot{v}(t) = \frac{Tu(t) - Av^2(t)e^{-\kappa\left(r(t)-r_0\right)}}{m(t)} - \frac{1}{r^2(t)} & t \in [0, t_{\mathrm{f}}] \\ \dot{m}(t) = -bu(t) & t \in [0, t_{\mathrm{f}}] \\ \big(r(0), v(0), m(0)\big) = (r_0, 0, m_0) \end{cases}$$

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 integrate the equations numerically by using fourth-order Runge-Kutta method, as we did in the previous example, where the control is approximated by means of piecewise constant functions. Before defining our stochastic optimization problem, we first show the solution to the deterministic one:

$$\begin{cases} \max\limits_{(t_{\mathrm{f}},u)\in\mathbb{R}_+\times\mathcal{U}} m(t_{\mathrm{f}}) \\ r(t_{\mathrm{f}}) \geq r_{\mathrm{f}}. \end{cases} \tag{4.7}$$

Table 6 sums up the choice of parameters for this model. The optimal final time and cost found by WORHP are $t_{\mathrm{f}}^* \approx 0.1742$ and $m(t_{\mathrm{f}}^*) \approx 0.6297$. Figure 5 shows the corresponding optimal trajectory.

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 6: Parameters for the deterministic optimization

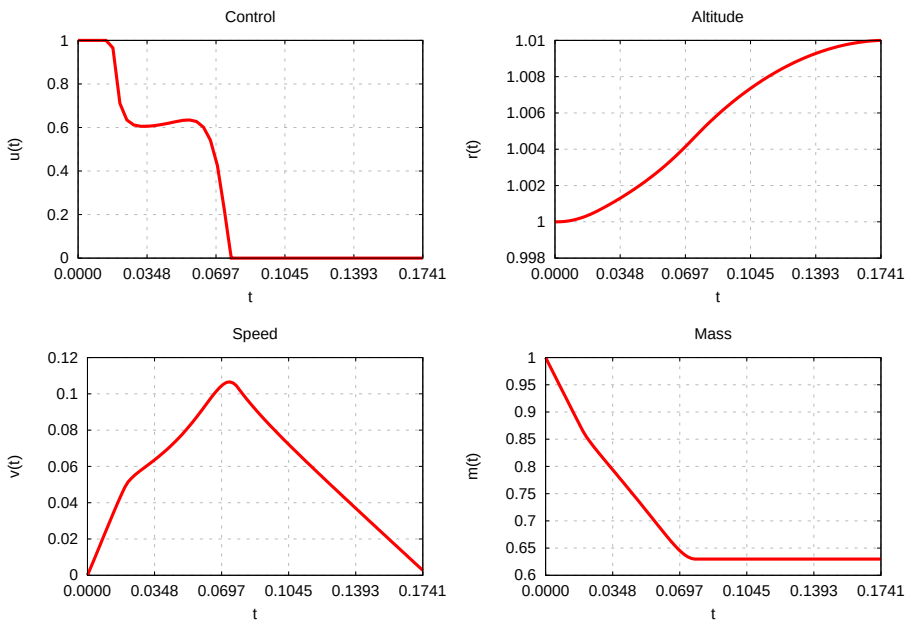


Figure 5: Plot of control, altitude, speed and mass for the Goddard problem.

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.

$$\mathbb{E}\left[m(t_\mathrm{f})\right] = \mathbb{E}\left[\int_0^{t_\mathrm{f}} m_0 - \frac{T}{v_\mathrm{e}} u(t)\,\mathrm{d}t\right] = \int_0^{t_\mathrm{f}} m_0 - \frac{\mathbb{E}[T]}{v_\mathrm{e}} u(t)\,\mathrm{d}t.$$

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

$$\overline{m}(t_\mathrm{f}) := \int_0^{t_\mathrm{f}} m_0 - \frac{\overline{T}}{v_\mathrm{e}} u(t)\,\mathrm{d}t.$$

This leads us to the stochastic optimization problem

$$\begin{cases} \max\limits_{(t_\mathrm{f},u)\in\mathbb{R}_+\times\mathcal{U}} \overline{m}(t_\mathrm{f}) \\ \mathbb{P}\big[R_\mathrm{f}(T, t_\mathrm{f}, u) \geq r_\mathrm{f}\big] \geq p \end{cases} \tag{4.8}$$

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 shows the choice of parameters defined in this subsection.

Parameter	p	$\overline{T}$	ΔT
Value	0.9	3.5	0.1

Table 7: Additional parameters for the stochastic optimization

Application of the method. By using the definition of the density function $f_{t_\mathrm{f},u}$ of the random variable $r_u(T, t_\mathrm{f})$, we can rewrite problem (4.8) as

$$\begin{cases} \max\limits_{(t_\mathrm{f},u)\in\mathbb{R}_+\times\mathcal{U}} \overline{m}(t_\mathrm{f}) \\ F_{(t_\mathrm{f},u)}(r_\mathrm{f}) \leq 1-p. \end{cases} \tag{4.9}$$

If we replace F_u with its KDE approximation $\hat{F}_u$, our problem becomes

$$\begin{cases} \max\limits_{(t_\mathrm{f},u)\in\mathbb{R}_+\times\mathcal{U}} \overline{m}(t_\mathrm{f}) \\ \hat{F}_{(t_\mathrm{f},u)}(r_\mathrm{f}) \leq 1-p. \end{cases} \tag{4.10}$$

The procedure used for solving problem (4.10) is described in 3.2, with the only difference that for this example we do not take a random sample from the variable T. Since we only have one random variable, we can take a uniform deterministic sample of T by dividing the interval $[T_-, T_+]$ into $n-1$ sub-intervals. We choose to use the SNR method (see 3.5) for computing the bandwidth combined with the Gaussian kernel.

Numerical results. Figures 6 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 instance, for $n = 500$ the optimal

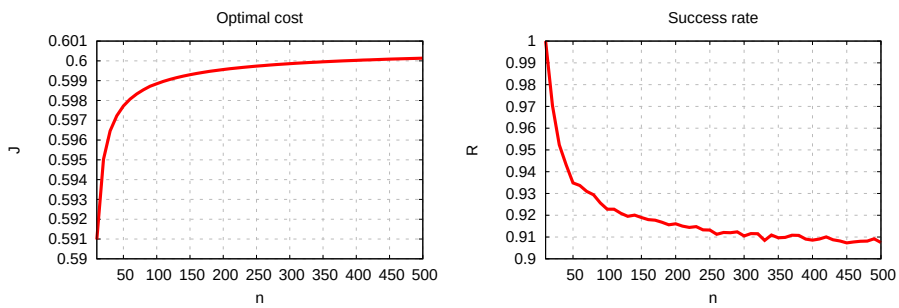


Figure 6: Plot of $m(t_\mathrm{f}, u^*)$ and R as functions of n.

final time is $t_\mathrm{f}^* \approx 0.1881$, with a corresponding cost $\overline{m}(t_\mathrm{f}^*) \approx 0.6001$ and a success rate $R = 90.81\%$. The corresponding optimal control u^* is shown in Figure 7. Figure 8 the other related plots. Table 8 and Figure 9 compare the solution we just found for the stochastic optimization problem to the two solutions we obtain from the deterministic one in the best and worst cases. It can be seen how the solution to the chance constrained problem is slightly better than the one in the worst case, but still lower than the one corresponding to the best case. Interestingly, Figure 9 shows that the shape of the control strategy does

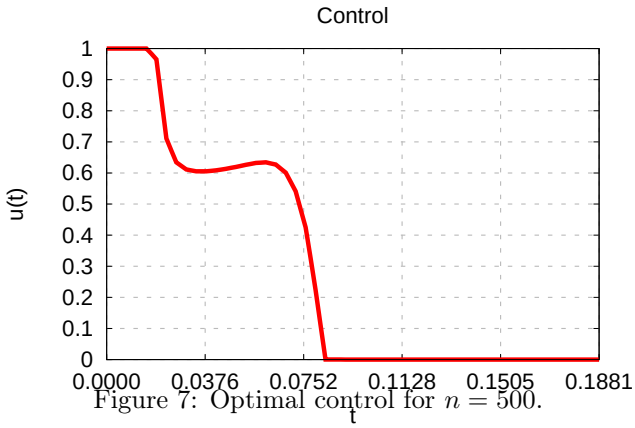


Figure 7: Optimal control for $n = 500$.

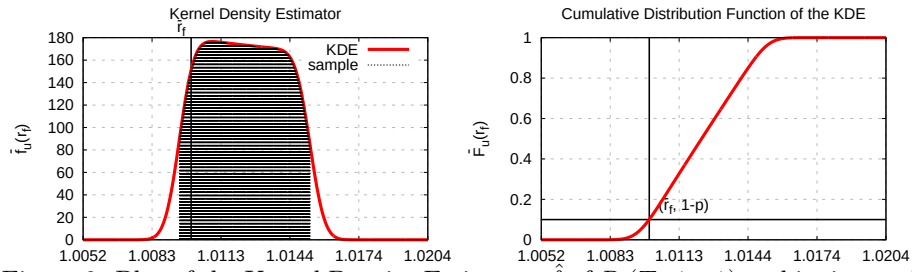


Figure 8: Plot of the Kernel Density Estimator $\hat{f}$ of $R_{\mathrm{f}}(T, t_{\mathrm{f}}^*, u^*)$ and its integral $\hat{F}$.

Case	T	t_{f}^*	$\overline{m}(t_{\mathrm{f}}^*)$
Random	$\sim U(T_-, T_+)$	**0.1881**	**0.6001**
Best	T_+	**0.1613**	**0.6584**
Worst	T_-	**0.1902**	**0.5928**

Table 8: Result comparison for extremal values of T.

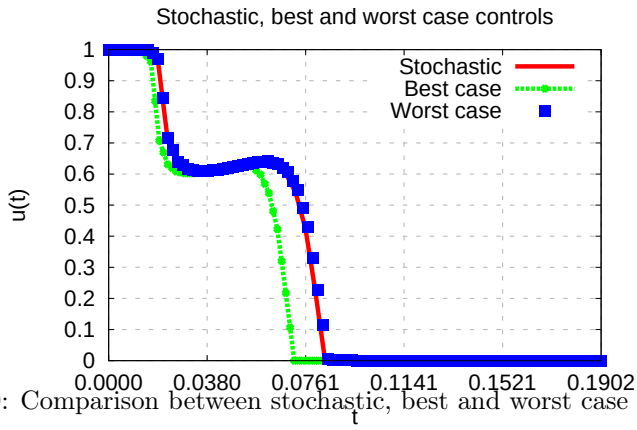


Figure 9: Comparison between stochastic, best and worst case controls.

not change much between the three cases, and the main difference lies in the

optimal value for the final time t_{f}^*. Table 9 shows the comparison between the solution of the deterministic problem (4.7) and its stochastic counterpart (4.8) 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 5).

n	p	ΔT	h	$\overline{m}(t_{\mathrm{f}}^*)$	R
Stochastic					
500	0.8	0.5	0.00517	**0.4808**	**0.7980**
		0.25	0.00155	**0.5701**	**0.8011**
		0.1	0.00048	**0.6085**	**0.7996**
		0.05	0.00022	**0.6197**	**0.7999**
		0.025	0.00010	**0.6267**	**0.6879**
	0.9	0.5	0.00813	**0.3866**	**0.9096**
		0.25	0.00185	**0.5419**	**0.9090**
		0.1	0.00051	**0.6001**	**0.9090**
		0.05	0.00023	**0.6162**	**0.8929**
		0.25	0.00011	**0.6222**	**0.9479**
	0.995	0.5	0.02127	**0.1538**	**1.0000**
		0.25	0.00271	**0.4728**	**1.0000**
		0.1	0.00057	**0.5827**	**1.0000**
		0.05	0.00024	**0.6075**	**1.0000**
		0.025	0.00011	**0.6186**	**1.0000**
Deterministic					
—				0.6297	—

Table 9: Result comparison for different values of n, p and ΔT.

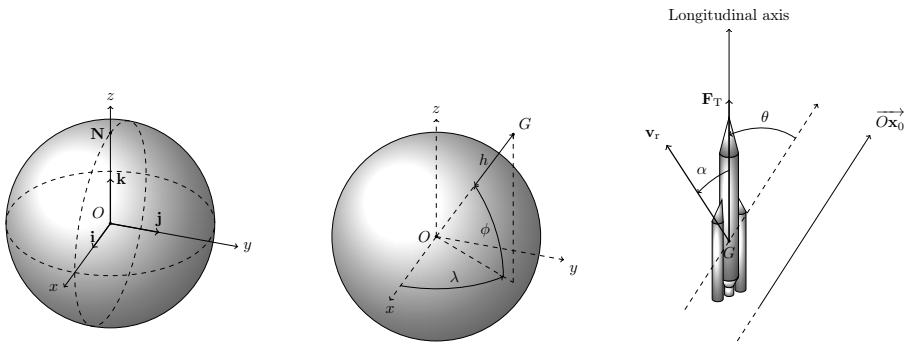
4.3 Complex three stage launcher with one decision variable and two random variables

We now study the more complex model of a real space launcher and define a percentile optimization problem in the form

$$\begin{cases} \min\limits_{\mu\in\mathbb{R}} \mu \\ \mathbb{P}\big[G(\xi) \leq \mu\big] \geq p \end{cases} \tag{4.11}$$

The name comes from the fact that this problem aims at finding the p-percentile μ of the distribution of $G(\xi)$. 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. We start with the frame of reference. We define the inertial equatorial frame coordinate system $\mathcal{S} := (O, \mathbf{i}, \mathbf{j}, \mathbf{k})$ in Figure 10(b), 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 and $\mathbf{j} := \mathbf{k} \times \mathbf{i}$ completes the coordinate system. In this



(a) The coordinates of G (b) The coordinate system $\mathcal{S}$ (c) The angles θ and α

Figure 10: Reference's frame.

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 10(a). Where ϕ is the latitude, λ is the longitude and h is the height. The conversion formulas between cartesian and geographic coordinates can be found in [11]. There is a number of variables and parameters attached to the launcher. We first define its longitudinal axis: This axis passes through G and points towards the edge of the launcher (see Figure 10(c)). 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 $\psi = \arcsin(\cos(i)/\cos(\phi_0))$, meaning 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. The orbit plane is the plane of the ellipse that defines the Geostationary Transfer Orbit (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.

Moreover, 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 $m_{\mathrm{e}i0} := m_{\mathrm{e}i}(t_0) \quad \forall i \in \{1,2,3\}$, and 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.

The launcher is subject to three forces: The force due to gravity, the drag force and the thrust force. The gravity force is

$$\mathbf{F}_{\mathrm{G}}(m,\mathbf{x}) = -\begin{pmatrix} F_{\mathrm{G}x}(m,\mathbf{x}) & 0 & 0 \\ 0 & F_{\mathrm{G}y}(m,\mathbf{x}) & 0 \\ 0 & 0 & F_{\mathrm{G}z}(m,\mathbf{x}) \end{pmatrix} \frac{\mathbf{x}}{||\mathbf{x}||}$$

where

$$\begin{aligned} F_{\mathrm{G}x}(m,\mathbf{x}) = F_{\mathrm{G}y}(m,\mathbf{x}) &= m\frac{\mu_0}{||\mathbf{x}||^2}\left(1 + J_2\frac{3}{2}\frac{R_{\mathrm{e}}^2}{||\mathbf{x}||^2}\left(1 - 5\frac{z^2}{||\mathbf{x}||^2}\right)\right) \\ F_{\mathrm{G}z}(m,\mathbf{x}) &= m\frac{\mu_0}{||\mathbf{x}||^2}\left(1 + J_2\frac{3}{2}\frac{R_{\mathrm{e}}^2}{||\mathbf{x}||^2}\left(3 - 5\frac{z^2}{||\mathbf{x}||^2}\right)\right) \end{aligned}$$

μ_0 is the standard gravitational parameter of the Earth and J_2 is the correction factor due to its oblateness. The drag force is

$$\mathbf{F}_{\mathrm{D}}(\mathbf{x},\mathbf{v}) = -F_{\mathrm{D}}(\mathbf{x},\mathbf{v})\frac{\mathbf{v}_{\mathrm{r}}(\mathbf{x},\mathbf{v})}{||\mathbf{v}_{\mathrm{r}}(\mathbf{x},\mathbf{v})||}$$

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}. The thrust force is

$$\mathbf{F}_{\mathrm{T}}(\theta,\mathbf{x},\mathbf{v}) = F_{\mathrm{T}}(\mathbf{x})\mathbf{i}_{\mathrm{T}}(\theta,\mathbf{x},\mathbf{v})$$

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. We can now write the equations of motion in cartesian coordinates:

$$\begin{cases} \dot{\mathbf{x}}(t) = \mathbf{v}(t) \\ m(t)\dot{\mathbf{v}}(t) = \mathbf{F}_{\mathrm{G}}\big(m(t), \mathbf{x}(t)\big) + \mathbf{F}_{\mathrm{D}}\big(\mathbf{x}(t), \mathbf{v}(t)\big) + \mathbf{F}_{\mathrm{T}}\big(\theta(t), \mathbf{x}(t), \mathbf{v}(t)\big) \\ \dot{m}(t) = -\beta. \end{cases} \tag{4.12}$$

We can control the direction of the launcher by acting on the pitch angle θ at any time t. For a given position $\mathbf{x}$ and velocity $\mathbf{v}$, the perigee and apogee of the associated orbit are given by

$$L_{\mathrm{p}}(\mathbf{x}, \mathbf{v}) = \big(1 - \epsilon(\mathbf{x}, \mathbf{v})\big)a(\mathbf{x}, \mathbf{v}) - R_{\mathrm{e}} \qquad L_{\mathrm{a}}(\mathbf{x}, \mathbf{v}) = \big(1 + \epsilon(\mathbf{x}, \mathbf{v})\big)a(\mathbf{x}, \mathbf{v}) - R_{\mathrm{e}}$$

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 $a(\mathbf{x}, \mathbf{v}) = \frac{1}{\frac{2}{||\mathbf{x}||} - \frac{||\mathbf{v}||^2}{\mu_0}}$. 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 $m(t) = m_{\mathrm{p}} + m_{\mathrm{c}} + m_{\mathrm{f}} + \sum_{i=1}^{3}(1 + K_i)m_{\mathrm{e}i}(t) \quad \forall t \in [t_0, t_1)$. 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. The pitch angle for this sub-phase is $\theta(t) \equiv 0 \quad \forall t \in [t_0, t_{1.1})$. Then the launcher rotates with constant speed changing its orientation:

$$\theta(t) = \frac{\theta_1}{\tau_{1.2}}(t - t_{1.1}) \quad \forall t \in [t_{1.1}, t_{1.2}).$$

After the tilt, direction of the thrust is fixed at the final values of the previous sub-phase until the angle of incidence α is zero:

$$\theta(t) = \theta_1 \quad \forall t \in [t_{1.2}, t_{1.3}),$$
$$\text{where} \quad t_{1.3} := \min_{t \in (t_{1.2}, +\infty)} \big\{ t \quad | \quad \alpha(t) = 0 \big\}.$$

The final sub-phase is a 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 $m(t) = m_{\mathrm{p}} + m_{\mathrm{c}} + m_{\mathrm{f}} + \sum_{i=2}^{3}(1 + K_i)m_{\mathrm{e}i}(t) \quad \forall t \in [t_1, t_{2.1})$. The second stage engine ignites. This sub-phase ends with the release of the fairing, as soon as the heat flux decreases to a given value:

$$\theta(t) = \theta_2 + \theta_2'(t - t_1) \quad \forall t \in [t_1, t_{2.1})$$
$$\text{where} \quad t_{2.1} := \min_{t \in (t_1, +\infty)} \Big\{ t \quad | \quad \Gamma\big(\mathbf{x}(t), \mathbf{v}(t)\big) \leq \Gamma^* \Big\}$$

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 $m(t) = m_{\mathrm{p}} + m_{\mathrm{c}} + \sum_{i=2}^{3}(1 + K_i)m_{\mathrm{e}i}(t) \quad \forall t \in [t_{2.1}, t_2)$. The flight continues

without fairing until complete consumption of the fuel in the second stage: $\tau_2 = \frac{m_{e20}}{\beta_2}$. This sub-phase ends with the jettison of the second stage, and the pitch angle is $\theta(t) = \theta_2 + \theta_2'\tau_{2.1} + \theta_2'(t - t_1) \quad \forall t \in [t_{2.1}, t_2)$.

- **Phase 3:** During this phase the mass of the launcher is $m(t) = m_p + m_c + (1 + K_3)m_{e3}(t) \quad \forall t \in [t_{2.2}, t_f)$. The third stage engine ignites, and this phase ends when the third stage's fuel is exhausted: $\tau_3 = \frac{m_{e30}}{\beta_3}$. At final time $t_f := t_3$ the the final position and velocity have to be compatible with the target orbit:

$$\begin{aligned} \theta(t) &= \theta_3 + \theta_3'(t - t_2) \quad \forall t \in [t_{2.2}, t_f) \\ L_p\big(\mathbf{x}(t_f), \mathbf{v}(t_f)\big) &= L_p^* \\ L_a\big(\mathbf{x}(t_f), \mathbf{v}(t_f)\big) &= L_a^*. \end{aligned}$$

We can now formulate the following deterministic optimization problem.

$$\begin{cases} \min\limits_{(m_{e30},\theta_1,\theta_2,\theta_2',\theta_3,\theta_3') \in \mathbb{R}_+^6} m_{e30} \\ L_p(m_{e30}, \theta_1, \theta_2, \theta_2', \theta_3, \theta_3') = L_p^* \\ L_a(m_{e30}, \theta_1, \theta_2, \theta_2', \theta_3, \theta_3') = L_a^* \end{cases} \tag{4.13}$$

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 $\big(\mathbf{x}(t_f), \mathbf{v}(t_f)\big)$, according to the decision variables $(m_{e30}, \theta_1, \theta_2, \theta_2', \theta_3, \theta_3')$.

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_{sp1}	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_{sp2}	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_{sp3}	450.72 s
S_3	1.97 m^2
A_3	17.35 m^2

Table 10: Mechanical and structural parameters

Table 10 summarizes the choice of all the fixed parameters of the problem while Figure 11 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 11) because of the relation $m_{ei0} = \beta_i\tau_i$ for $i \in \{1, 2, 3\}$. The parameters for the Earth and the flight sequence are defined in Tables 12 and 13 respectively. The

m_{e10}	278797.26 kg
m_{e20}	60714.78 kg

Table 11: Values for the initial fuel masses

optimal values found by WORHP for the optimization variables are reported in Table 14 and Figure 12 shows the corresponding optimal trajectory. The ODE system (4.12) is integrated by using the Fortran 90 subroutine DOP853 described in [14].

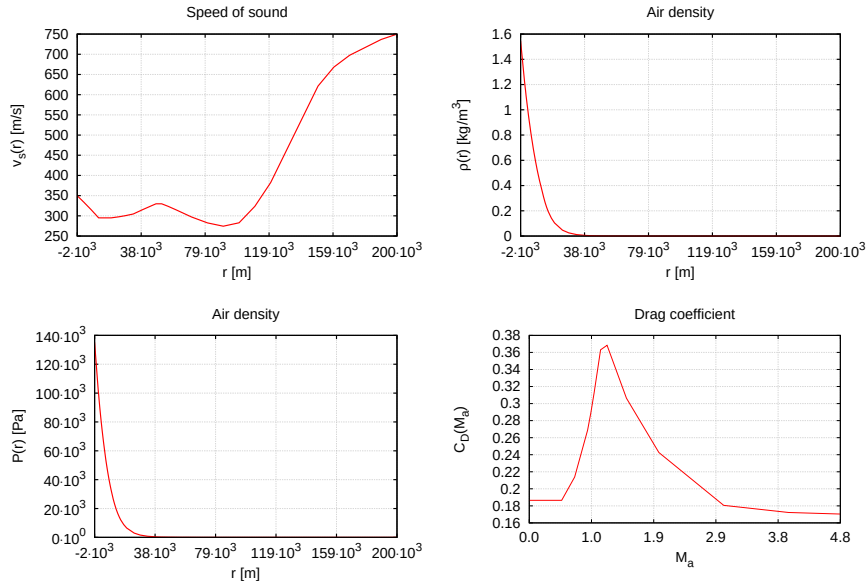


Figure 11: Speed of sound v_{s}, air density ρ, atmospheric pressure P and drag coefficient C_{D}.

Ω	$7.292155{\cdot}10^{-5}$ rad/s
R_{p}	6356752 m
R_{e}	6378137 m
μ_0	$3.986005{\cdot}10^{14}$ m^3/s^2
J_2	$1.08263{\cdot}10^{-3}$
g_0	9.80665 m/s^2

Table 12: Earth's parameters

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 13: Parameters for the flight sequence

Problem statement. Let $M_{\mathrm{e}30}(\pi, m_{\mathrm{p}})$ be the value function of (4.13), depending on $\pi := (I_{\mathrm{sp}3}, K_3)$ and the dimensioning parameter m_{p}. Consider the

$m_{\mathrm{e}30}$	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 14: Optimal values for the free variables

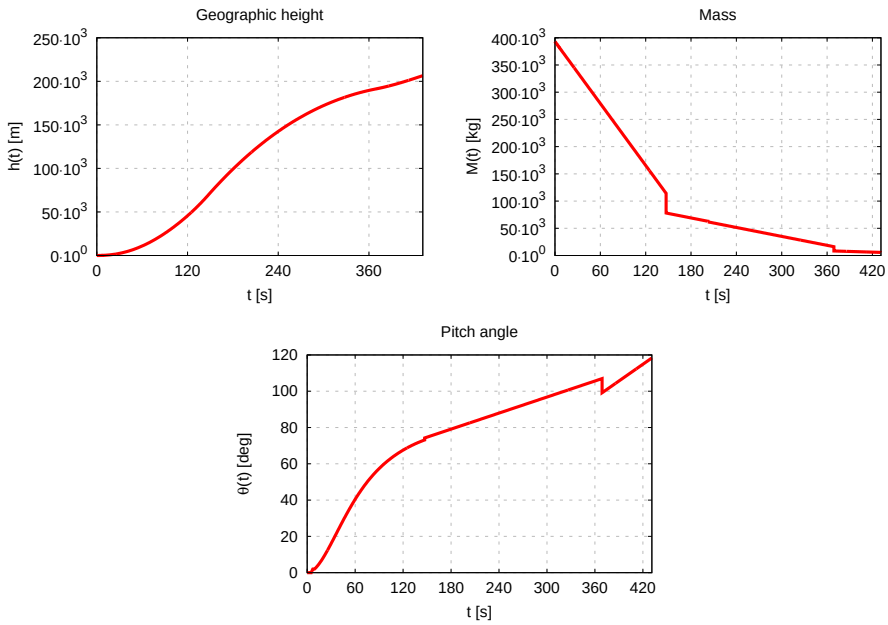


Figure 12: Result of the three-stage launcher optimization

following chance constrained optimization problem

$$\begin{cases} \min\limits_{\mu \in \mathbb{R}_+} \mu \\ \mathbb{P}\big[M_{\mathrm{e}30}(\pi, m_{\mathrm{p}}) \geq \mu\big] \geq p \end{cases} \tag{4.14}$$

where $I_{\mathrm{sp}3}$ 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 $\overline{K}_3$:

$$I_{\mathrm{sp}3} \sim U(I_{\mathrm{sp}3-}, I_{\mathrm{sp}3+}) \qquad K_3 \sim U(K_{3-}, K_{3+}).$$

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 (4.14) matches the definition of the percentile optimization problem (4.11). We remark that this problem, and thus its solution, depends on two dimensioning parameters: The payload m_{p} and the probability of success p. Table 15 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

Parameter	p	$\overline{I}_{\mathrm{sp}3}$	$\Delta I_{\mathrm{sp}3}$	$\overline{K}_3$	ΔK_3
Value	0.9	450.72 [s]	0.1	0.13	0.1

Table 15: Additional parameters for the stochastic optimization

the chance constraint in the general form $\mathbb{P}\big[G(x,\xi) \leq 0\big] \geq p$. In the particular case of this example's model, the inequality above can be rewritten as $\mathbb{P}\big[D(\pi) \leq E(\mu)\big] \geq p$, allowing us to improve the solver's performances by pre-computing the function $M_{\mathrm{e}30}(\pi, m_{\mathrm{p}})$ 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 13 shows the plot of $M_{\mathrm{e}30}(\pi, m_{\mathrm{p}})$ as a function of π for our choice of m_{p} (see Table 10). The function has been evaluated at 16 values of π on an equally partitioned grid on the set $[I_{\mathrm{sp}3-}, I_{\mathrm{sp}3+}] \times [K_{3-}, K_{3+}]$. 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 require a new computation of $M_{\mathrm{e}30}$ at grid values. For all the values of π in $[I_{\mathrm{sp}3-}, I_{\mathrm{sp}3+}] \times [K_{3-}, K_{3+}]$ the solver WORHP

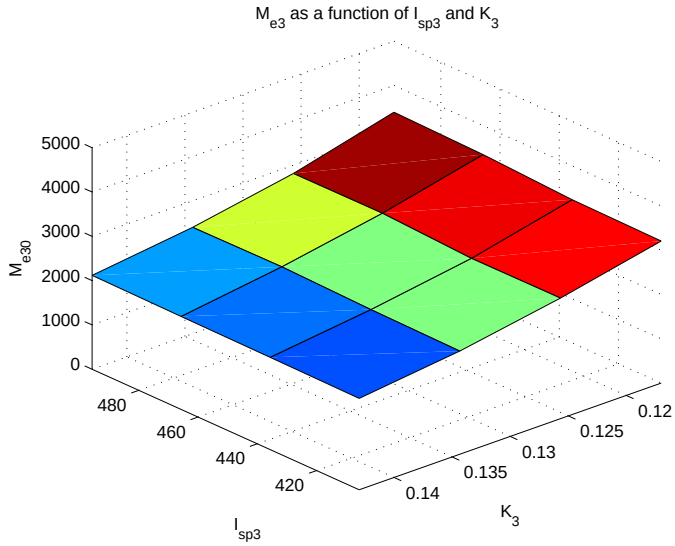


Figure 13: Plot of the third stage optimal fuel mass as a function of π.

was able to compute an optimal control allowing the launcher to reach its final orbit while minimizing the initial mass.

Application of the method. 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 (4.14) as

$$\begin{cases} \min\limits_{\mu \in \mathbb{R}_+} \mu \\ F_{m_{\mathrm{p}}}(\mu) \geq 1 - p. \end{cases} \tag{4.15}$$

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 $F_{m_{\mathrm{p}}}$ via KDE by drawing a sample of size from the array of random variables π. Our problem becomes

$$\begin{cases} \min\limits_{\mu \in \mathbb{R}_+} \mu \\ \hat{F}_{m_{\mathrm{p}}}(\mu) \geq 1 - p. \end{cases} \tag{4.16}$$

The procedure used for solving problem (4.16) is described in 3.2. We choose to use the SNR method (see 3.5) for computing the bandwidth combined with the Gaussian kernel.

Numerical results. Figures 14 to 15 show the behavior of ten sequences 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

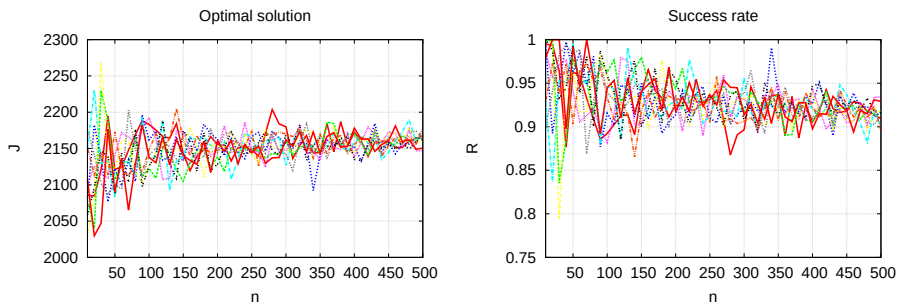


Figure 14: Plot of μ^* and R as functions of n (ten simulations).

optimal cost and the success rate are $\mu^* \approx 2162.78$ and $R \approx 91.83\%$. Figure 16 shows the related plots.

5 Conclusion

The performances of KDE method (coupled with a nonlinear problem solver) may vary depending on a variety of factors. The nature of the problem, first: Whether the decision variables and controls are separable from the random variables or not has a strong impact on the method, both from the theoretical and numerical point of view. The bandwidth selection strategy also plays an important role: Some of the most refined methods to compute the bandwidth might require the minimization of an error function. The quadrature formula used for the numerical integration of the density estimator, the discretization as an optimization problem, and the choice of the optimization solver itself strongly influence the results.

Throughout this paper we showed how chance constrained optimization can be relevant to solve 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

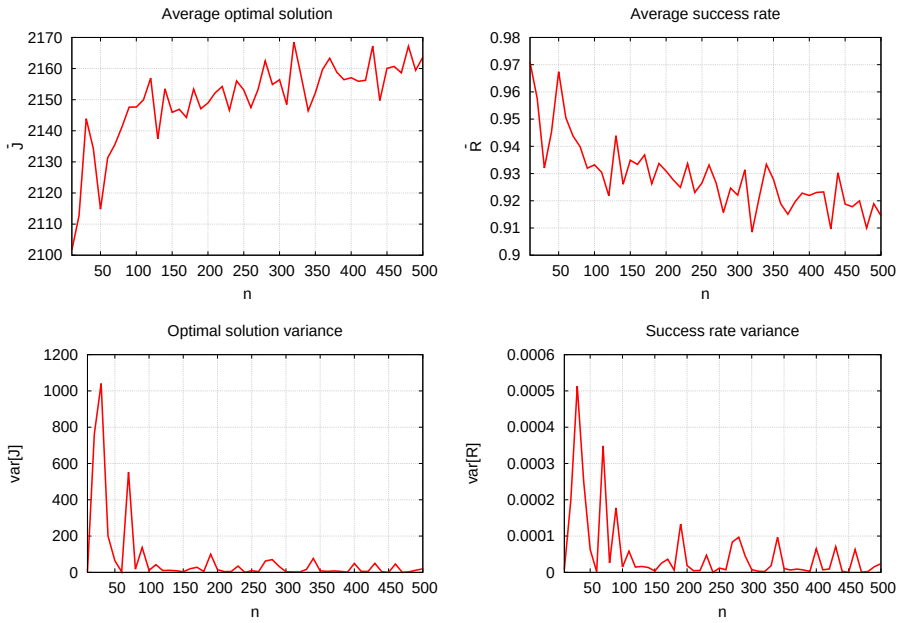


Figure 15: Plot of the average value and variance of μ^* and R as functions of n.

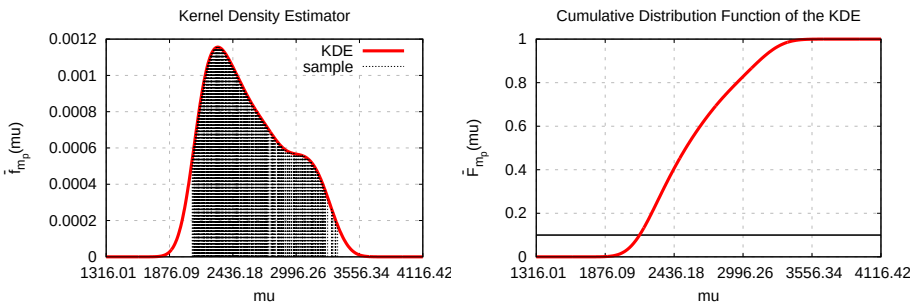


Figure 16: Plot of the Kernel Density Estimator $\hat{f}_{m_\mathrm{p}}$ of $M_{\mathrm{e}30}$ and its integral $\hat{F}$.

solutions. In spite of a not yet complete theoretical framework, the numerical results provided by Kernel Density Estimation are very promising. Even better results might be obtained by improving the computation of the bandwidth h, for example, by substituting the second derivative f'' of the unknown density in (3.3) with some approximation (this so-called *plug-in* method is explained in detail in [32]). Such a method can increase the accuracy of the estimator $\hat{f}$, but as it involves more complex operations for the computation of h compared to the Simple Normal Reference bandwidth (3.5), we decided to implement the latter in our tests in order to preserve good performances. Regardless of the particular implementation of KDE, pairing it with a robust NLP solver like WORHP has proven solid enough to handle the chance constrained optimization problems in our examples. It is our hope that this paper will foster future research along this line.

References

[1] I.A. Ahmad and M. Amezziane. A general and fast convergent bandwidth selection method of kernel estimator. *Journal of Nonparametric Statistics*, 19(4-5):165–187, 2007.

[2] L. Andrieu, G. Cohen, and F. J. Vásquez-Abad. Stochastic programming with probability constraints. *arXiv*, 0708.0281, 2007.

[3] D. Berstimas. The price of robustness. *Operations Research*, 52(1):35–53, 2004.

[4] P. K. Bhattacharya. Estimation of a probability density function and its derivatives. *Sankhyā: The Indian Journal of Statistics*, 29(4):373–382, 1967.

[5] 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.

[6] G. C. Calafiore and M. C. Campi. The scenario approach to robust control design. *IEEE Transactions on Automatic Control*, 51(5):742–753, 2006.

[7] B. A. Calfa, I. E. Grossmann, A. Agarwal, S. J. Bury, and J. M. Wassick. Data-driven individual and joint chance-constrained optimization via kernel smoothing. *Computers & Chemical Engineering*, 78:51–69, 2015.

[8] 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.

[9] D. Dentcheva. Optimization models with probabilistic constraints. In F. Dabbene G. Calafiore, editor, *Probabilistic and Randomized Methods for Design under Uncertainty*. Springer, 2003.

[10] L. Devroye and L. Györfi. *Nonparametric Density Estimation: the L_1 view*. Wiley, 1985.

[11] 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.

[12] E. Giné and A. Guillou. Rates of strong uniform consistency for multivariate kernel density estimators. *Annales de l'Institut Henri Poincare: Probability and Statistics*, 38(6):907–921, 2002.

[13] R. H. Goddard. A method of reaching extreme altitudes. *Smithsonian Miscellaneous Collections*, 71(2):2–69, 1921.

[14] 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.

[15] R. Henrion and W. Römisch. Hölder and lipschitz stability of solution sets in programs with probabilistic constraints. *Mathematical Programming*, 100(3):589–611, 2004.

[16] J. L. Hodges and E. L. Lehmann. The efficiency of some nonparametric competitors of the t-test. *The Annals of Mathematical Statistics*, 27(2):324–335, 1956.

[17] K. Marti. Differentiation formulas for probability functions: The transformation method. *Mathematical Programming*, 75(2):201–220, 1996.

[18] M. C. Minnotte. Achieving higher-order convergence rates for density estimation with binned data. *Journal of the American Statistical Association*, 93(442):663–672, 1998.

[19] É. A. Nadaraya. On non-parametric estimates of density functions and regression curves. *Theory of Probability & Its Applications*, 10(1):186–190, 1965.

[20] A. Nemirovski and A. Shapiro. Convex approximations of chance constrained programs. *SIAM Journal on Optimization*, 17(4):969–996, 2006.

[21] E. Parzen. On estimation of a probability density function and mode. *The Annals of Mathematical Statistics*, 33(3):1065–1076, 1962.

[22] B. L. S. Prakasa Rao. *Non-Parametric Functional Estimation.* Academic Press, 1983.

[23] A. Prékopa. On probabilistic constrained programming. *Proceedings of the Princeton Symposium on Mathematical Programming*, pages 113–138, 1970.

[24] A. Prékopa. Contributions to the theory of stochastic programming. *Mathematical Programming*, 4(1):202–221, 1973.

[25] A. Prékopa. *Stochastic programming.* Kluwer Academic Publishers, 1995.

[26] A. Prékopa. Probabilistic programming. In A. Shapiro A. Ruszczuǹski, editor, *Stochastic programming*, volume 10. Elsevier, 2003.

[27] E. Raik. Qualitative research into the stochastic nonlinear programming problems. *Eesti NSV Teaduste Akademia Toimetised*, 20:8–14, 1971.

[28] M. Rosenblatt. Remarks on some nonparametric estimates of a density function. *The Annals of Mathematical Statistics*, 27(3):832–837, 1956.

[29] K. H. Sahin and U. M. Diwekar. Better optimization of nonlinear uncertain systems (bonus): A new algorithm for stochastic programming using reweighting through kernel density estimation. *Annals of Operations Research*, 132(1):47–68, 2004.

[30] 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.

[31] 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.

[32] S. J. Sheather. Density estimation. *Statistical Science*, 19(4):588–597, 2004.

[33] B. W. Silverman. Weak and strong uniform consistency of the kernel estimate of a density and its derivatives. *The Annals of Statistics*, 6(1):177–184, 1978.

[34] B. W. Silverman. *Density Estimation for Statistics and Data Analysis*. Chapman & Hall, 1986.

[35] G. R. Terrell and D. W. Scott. Variable kernel density estimation. *The Annals of Statistics*, 20(3):1236–1265, 1991.

[36] S. Uryasev. Derivatives of probability functions and some applications. *Annals of Operations Research*, 56(1):287–311, 1995.

[37] 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.

[38] R. B. Vinter. *Optimal Control*. Springer, 2000.

[39] A. Wald. Contributions to the theory of statistical estimation and testing hypotheses. *The Annals of Mathematics*, 10(4):299–326, 1939.

[40] D. M. Young and R. T. Gregory. *A survey of numerical mathematics*. Addison-Wesley, 1972.

[41] Y. Zhang, Y. Feng, and G. Rong. Data-driven chance constrained and robust optimization under matrix uncertainty. *Industrial & Engineering Chemistry Research*, 55(21):6145–6160, 2016.