DISSERTATION

Titel der Dissertation

“Uncertainty modeling in higher dimensions:
Towards robust design optimization”

Verfasser

Martin Fuchs

angestrebter akademischer Grad

Doktor der Naturwissenschaften (Dr.rer.nat)

Wien, 27. August 2008

Studienkennzahl lt. Studienblatt:	A 091 405
Dissertationsgebiet lt. Studienblatt:	Mathematik
Betreuer:	o. Prof. Dr. Arnold Neumaier

Uncertainty modeling in higher dimensions: Towards robust design optimization

by

Martin Fuchs

Uncertainty modeling in higher dimensions: Towards robust design optimization

Abstract

Modern design problems impose multiple major tasks an engineer has to accomplish.

- The design should account for the designated functionalities.
- It should be optimal with respect to a given design objective.
- Ultimately the design must be safeguarded against uncertain perturbations which should not cause failure of the design.

These tasks are united in the problem of robust design optimization giving rise to the development of computational methods for uncertainty modeling and design optimization, simultaneously.

Methods for uncertainty modeling face some fundamental challenges: The computational effort should not exceed certain limitations; unjustified assumptions must be avoided as far as possible. However, the most critical issues concern the handling of incomplete information and of high dimensionality. While the low dimensional case is well studied and several methods exist to handle incomplete information, in higher dimensions there are only very few techniques. Imprecision and lack of sufficient information cause severe difficulties – but the situation is not hopeless.

In this dissertation, it is shown how to transfer the high-dimensional to the one-dimensional case by means of the potential clouds formalism. Using a potential function, this enables a worst-case analysis on confidence regions of relevant scenarios. The confidence regions are weaved into an optimization problem formulation for robust design as safety constraints. Thus an interaction between optimization phase and worst-case analysis is modeled which permits a posteriori adaptive information updating. Finally, we apply our approach in two case studies in 24 and 34 dimensions, respectively.

Preface

It has been a long journey to the completion of this dissertation, taking a few detours, but keeping in mind the goal which I have achieved today. However, in retrospect I now realize what is meant by: the journey is the reward.

I would like to acknowledge the help of the following people who supported me during this journey in various ways and thus contributed to this dissertation.

First of all, I would like to thank my supervisor, Prof. Arnold Neumaier, for his constant support, encouragement and enthusiasm, and for sharing his seemingly unlimited amount of inspiring ideas.

A very important support during my whole life – a subset of which is my time as a Ph.D. student – are my family and friends, in particular my parents, Christel and Ernst. Especially I want to thank my grandmother Hannelore who taught me basic arithmetics and many other things. I gratefully appreciate that and I miss you.

I would like to thank my diploma thesis supervisor Prof. Franz Hofbauer for teaching me the probability theoretical background of this thesis.

Thanks also to my former high school mathematics teacher Dieter Kauzleben whose lectures motivated me to study mathematics.

I would like to thank my very nice colleagues and friends from the Computational Mathematics Group and the Faculty of Mathematics for comments and discussions: Erich Dolejsi, Ferenc Domes, Mihaly Markot, Prof. Hermann Schichl, Harald Schilly, Peter Schodl, Prof. Oleg Shcherbina, and Dennis Westra.

Thanks to Prof. Christian Krattenthaler and our Dean Prof. Harald

Rindler for their help concerning some administrative and organizational issues.

I would like to thank Richard Welke for the "LaTeX-Inside" newsletter, and Lukas Grumet for his web design support.

Also I would like to thank my colleagues from the University of Szeged for the successful collaboration in our spacecraft design project: Tibor Csendes, Jozsef Dombi, Balazs Banhelyi, and Zsolt Gera.

As a great host during my time in Savannah I would like to thank Rafi Muhanna from Georgia Tech.

Last but not least I would like to thank all the nice people I met during my work at ESA: Dario Izzo, Tamas Vinko, Daniela Girimonte, Tobias Seidl, and many more.

Martin Fuchs

University of Vienna
August 2008

Contents

Chapter 1

Background

Modern technologies increasingly employ computational mathematics as a fundamental part of engineering activities. For example, the construction of a vehicle or a building requires the design to satisfy constraints on the **functionality** and **reliability** of the designed object. In other words, the object should serve the purpose planned by the engineer and should continue to do so also under uncertain, adverse conditions that may show up during utilization. Hence the engineer faces the task to qualify his creation as robust, i.e., to be safeguarded against worst-case scenarios. To this end the engineer takes into account uncertain perturbations in his calculations by way of specialized engineering computing methods.

Where do uncertainties originate from? Uncertainty can be imposed by measurements, especially cutting edge measurements. Here the calibration of the measurement device is the critical issue. Uncertainty is involved when estimating or predicting loads and stress on a construction. These are time variant uncertainties emerging as a structure faces several modifications during its lifetime. Changes in its usage, maintenance, material change due to chemical reactions all affect the structure to an unknown extend.

Whenever humans are involved in the production or employment of the designed object human errors can occur and give rise to uncertain scenarios. Misuse by non-expert users, or manufacturing errors in some components of the object may cause partial or complete failure of the design. Manufacturing errors can also be uncontrollable due to variability in the performance of production devices.

Uncertainty can arise from a lack of statistical data, e.g., if only very few or no measurements at all exist, as may happen in early design phases. Uncertainties can be provided together with stochastic information, such as distribution functions with parameters which are not precisely known, or moments of random variables. Expert knowledge sometimes is affected by uncertainties, in particular in case that no formal description can be given. Unformalized statements show up, e.g., when an expert cannot specify correlations exactly, but can formulate qualitative statements, like 'If the quantity a has a large value then quantity b cannot have a low value'. Thus he is able to exclude irrelevant scenarios, although he is unable to give a formal description of his knowledge.

Another source of uncertainty is model uncertainty. Engineers often use computational input-output models for subsystems of their design. These models may comprise idealizations, assumptions or simplifications. This leads to imprecision and uncertainty in the model output which must be taken into account for a reliable usage of the computational model.

In real-life design problems engineers have to deal with many of these uncertainties. One distinguishes two generic types of uncertainty. If the uncertainty is irreducible, i.e., if the randomness of uncertain scenarios is system inherent and physically uncontrollable (we mentioned variability of performance of production devices as an example), one speaks of **aleatory** uncertainty. If the uncertainty is reducible, i.e., if it arises from incomplete information or expert knowledge, one classifies the uncertainty as **epistemic**.

1.1 Uncertainty handling

Computational methods to handle the uncertainties should enable one to perform worst-case analysis on the basis of the available information. The **admissible failure probability**, i.e., the probability that a worst-case event causes a design failure, is supposed to be chosen with respect to failure costs. A rational decision about the design includes a reasonable estimation of failure costs consistent with the value system of the decision maker, not necessarily linearly associated with money. Admissible failure probabilities in case of a nuclear power plant, for example, should be significantly lower than for an ink-jet printer in some office. Modern technology frequently requires very low failure probabilities. Having fixed an admissible failure probability one seeks to determine computationally whether a given design fails at most with the allowed failure probability. If not the design is rejected.

The advantage of computational mathematics is that a method (e.g., to determine failure probabilities) formulated on a mathematical basis is applicable to a whole class of design problems and is developed to provide consistent results, i.e., equivalent problems lead to the same worst-case analysis. Ambiguity of the results, however, may be caused by unformalized, subjective or incomplete information, and may compromise consistency to a certain degree. Depending on the input information available for a design problem one identifies the problem class and applies a suitable method for that class. It is just like choosing the appropriate tool from a toolbox. This point of view can be described as a **toolbox philosophy**, cf. Nguyen [120]. This strategy to solve a problem is defined by the problem itself and the characteristics of the uncertainties involved. Thus different approaches to uncertainty modeling do not contradict each other, but rather constitute a mutually complementing framework.

There are some important questions which define a problem class and should be considered for each method of uncertainty handling: Which assumptions have been made? Are the assumptions justified?

For example, one may assume stationarity for random processes, or the Markov property. One may assume existence of a measure, say, in form of a given distribution function. One may assume independence. One may assume a uniform distribution for a random variable which is known to lie in a given interval, or a normal distribution for random variables where the distribution is actually unknown and only mean and variance are available.

In real-life applications many assumptions are difficult to justify rationally, with possibly drastic consequences. It can be shown that the normal distribution assumption can lead to critical underestimation of failure probabilities which can be bounded rigorously from a generalized Chebyshev inequality, cf. Section 2.1.3.

This leads to the next important question: the reliability and assessment of the results from real applications of the methods. The results can be **rigorous**, i.e., verified on the basis of a mathematical proof. They can be **approximate**, i.e., non-rigorous, but known to be close enough to the correct result. They can be **heuristic**, i.e., no guarantee for correct results and no rigorous error estimation can be given.

Another criterion for methods of uncertainty modeling is the computational effort. Sometimes computations are so expensive that they exceed any limit on time an engineer can afford to wait for a result. Hence the related methods are not applicable to any real-life problems or only to a very small class of problems.

A further question related to uncertainty methods is the information type that can be handled with the method. Information can be provided, e.g., as statistical data, distribution functions, expert knowledge etc.. Some methods reduce the information given, marginal distribution functions are simplified to intervals. This entails a loss of valuable information which would actually be available, but is not at all involved in the uncertainty model. On the contrary some methods estimate joint distributions from scarce statistical data, thus suggesting more information than actually exists. Closely related is the ultimate question about the methods, namely, how they treat imprecise or incomplete information. How does the quality of the method respond to a **lack of information** and to the **dimensionality** of the problem to solve? The dimension of a problem is determined by the number of uncertain variables involved. In some real-life design problems the dimension is low (say, smaller than about 5). In many problems, however, the dimension is significantly higher.

The low dimensional case is well studied. Reliable methods exist, originating from traditional modeling of uncertainty with classical probability theory. Issues with computational effort are almost absent and caused only by the complexity of the underlying system model of a specific problem rather than by the uncertainty model. Many methods start from the well known concept of only one-dimensional probability distributions or densities. This enables the straightforward computation of safety factors, cf. Section 2.1.5, or more elaborate approaches like reliability methods [31], [134]. Unfortunately in methods using classical probability theory the precise knowledge of distribution function is essential, so they turn out to be very sensitive to a lack of information.

Lack of information causes problems in probabilistic computations for some simple reasons. For instance, suppose we are given two uncertain quantities a and b. The available information comes from an expert who tells us that a is uniformly distributed in the interval $[1,2]$ and that b is a standard normally distributed random variable. Even in this very simple situation we cannot determine the distribution of $c := a + b$ as we do not know the joint distribution of a and b, but only their marginal distributions. Finding bounds on the distribution of functions of random variables, or bounds on failure probabilities, given imprecise or incomplete information supplemented by expert statements is a current research field, also in low dimensions. In this field the possibly most prominent methods are p-boxes, cf. Berleant & Cheng [10], Ferson [45], Dempster-Shafer theory, cf. Dempster [28], Shafer [144], and fuzzy sets, cf. Dubois & Prade [34], Zadeh [169]. Powerful statistical tools, like Kolmogorov-Smirnov bounds on empirical distributions [82], are available to develop advanced methodologies in low dimensions.

The high-dimensional case – appearing frequently in real-life applications – comprises severe difficulties. High-dimensionality can cause computations to become very expensive, with an effort growing exponentially with the dimension in many cases. This phenomenon is famous as the **curse of dimensionality**, cf., e.g., Koch et al. [80]. Even given the full knowledge about a joint distribution the numerical computation of error probabilities may be very time consuming, if not impossible. Moreover, rigorous computation or (preferably tight) bounding of failure probabilities can only be done in very few cases because the space of possible scenarios is too large. In higher dimensions full probabilistic models need to estimate high-dimensional distributions for which rarely sufficient data are available. Frequently it is just the other way around. Especially in early design phases data are scarce and the lack of information cannot be handled as reliable as in the low dimensional case (e.g., with Kolmogorov-Smirnov bounds on empirical distributions). Collecting a sufficient amount of statistical data for a formal description of uncertainty for high-dimensional problems (e.g., joint distributions or correlation information) is very difficult and rarely possible in practice. Sometimes some information can be obtained for 1D or 2D projections of the higher dimensional probability space. For example, information can be available in form of safety margins, i.e., intervals on single uncertain variables chosen without any formal knowledge, merely based on experience of experts. Or marginal distributions are given without correlations, but maybe with some vague dependency constraints.

Only few approaches exist that tackle the dimensionality issue and problems with incomplete information at the same time. These techniques can be based on sensitivity analysis methods towards a dimension reduction, cf. Oberguggenberger et al. [125], or on optimization over level sets of membership functions of fuzzy sets, cf. Möller et al. [105], or more generally on optimization over convex uncertainty regions, cf. Ben-Haim & Elishakoff [8]. More popular, but less reliable techniques are based on standard simulation methods which apply even in very high dimensions, as the computational effort they require is independent of the dimension. Unfortunately simulation techniques like Monte Carlo require a large amount of data (actually, complete knowledge of the probability distribution) and/or runtime to be reliable. Otherwise, unjustified assumptions on the uncertainties have to be made. The number of simulation runs required to safeguard against very small failure probabilities is known but high, cf. ter Marten et al. [155], in case of high-dimensionality. Lack of information endangers standard simulation based methods to critically underestimate the effects of the uncertain tails of probability distributions and hence failure probabilities, cf. Ferson [44]. Thus these techniques may fail to protect the design from failures of low probability. In many real-life applications the lack of data simply leads to a complete disregard of any rigorous methods. Often it is argued that the optimal design does not change whether one uses sophisticated uncertainty methods, or whether one uses subjective assumptions about safety margins by experts, cf. Sexsmith [143]. This may be a good argument for many examples. But of course, in consequence, no understanding and awareness of the sensitivity of the optimal design to the uncertainties can be developed by the specialists – and the **danger of drastically underestimating worst-cases** is always imminent.

We see on the one hand that the research field of uncertainty modeling in high dimensions given incomplete information has not yet matured very much – a lot of research still has to be dedicated to that field. On the other hand, however, the same situation in one dimension is already well-understood. We will follow the intuitive idea to transfer the higher dimensional case to the univariate case via the concept of **potential clouds** [59], [60], [116]. According to the toolbox philosophy mentioned previously we endeavor to develop a tool that can be applied to high-dimensional problems lacking sufficient stochastic information towards robustness in design optimization. We want to proceed in a computationally attractive fashion, reducing calculations to tractable global optimization and constraint satisfaction problems. We should be able to incorporate the information which is typically available in real-life applications. As we remarked, this includes safety margins, marginal distributions, sample data, and knowledge of experts who can provide formal and non-formal statements, and iterative information updates from assessing interim results. Many uncertainty models can only handle a certain type of information, but we will try to integrate in our methods as much of the available information as possible. Finally we wish to present a reliable and tractable worst-case analysis for the given high-dimensional information. The clouds formalism will give us means to determine a nested collection of confidence regions parameterized by the confidence level α. Thus we provide an instrument to support an engineer in the validation of a design, when he has to qualify the design as robust at the chosen confidence level.

1.2 Design optimization

In many design processes, robustness and design optimization are two separate steps. First, an optimal design is elaborated, e.g., fixing uncertain variables at their nominal values and optimizing given design objectives like minimal cost or minimal mass subject to given design constraints. Second, the design point is investigated for robustness by worst-case analysis with some suitable uncertainty model. The first step requires an algorithm that finds an optimal design point autonomously, i.e., merely computationally without human intervention.

For example, let us consider the case of spacecraft design. The design of a spacecraft is a demanding challenge. Several different engineering fields are addressed. This fact is summarized in the term **multidisciplinarity**. The multidisciplinarity and general complexity of spacecraft design make it difficult to obtain a complete survey and a deep understanding of the whole design process. An interaction between the involved disciplines is necessary. The resulting overall optimization is known as multidisciplinary design optimization (MDO), cf. Alexandrov & Hussaini [1]. MDO benefits particularly from optimization methods that computationally bridge the gap between different technical backgrounds.

Design optimization imposes optimization problem formulations of most complex nature, like **mixed integer nonlinear programming** (MINLP) with strong nonlinearities and even discontinuities. Additionally, since the people who implement the underlying model for the design are often working independently from the optimization phase, we have to cope with black box optimization. In view of all these difficulties standard optimization techniques do not apply. There exist no alternatives to heuristic methods yet. Some make use of a surrogate model, e.g., [37], [72], i.e., simplification to a similar problem which is easier to optimize. Some make use of enhanced search space sampling strategies, like genetic algorithms [65]. These methods will not be discussed in detail in this study.

Robust design optimization is able to perform both the worst-case analysis and the optimization phase in a single wrap. The combination of both leads to a **bilevel formulation** 'finding the best worst-case'. Sometimes the inner level, i.e., the worst-case search, is also called **anti-optimization** by Elishakoff [38].

For instance, remember MDO. In early design phases (very little information available) there exists a straightforward approach to integrate robustness directly into the optimization phase by concurrent engineering. Experts first assign safety margins, and then decide iteratively to refine or coarsen the margins on the basis of their judgment while converging to an optimal design. A certain value in between the safety margins which is deemed most worst-case relevant is fixed for each variable. These values are propagated within the whole optimization process. Thus the design arising from this process is supposed to include robustness intrinsically. Note that the assessment of robustness is exclusively based on expert knowledge of the engineers who define and refine the safety margins. The anti-optimization is done simply by assigning a fixed value to the uncertain variables. There is no quantification of reliability, no formal worst-case analysis involved.

Robust design optimization can also be done in different, less subjective ways. Reliability based design optimization (RBDO) incorporates the uncertainty model of probabilistic reliability methods into the optimization problem, cf. [78], [108]. Furthermore, design optimization can be based on possibility theory (PBDO) [109], evidence theory (EBDO) [110], or convex modeling [38]. Sometimes these methodologies allow for updating of information, or they attempt to capture the reasoning of the system experts, thus imitating design specialists' working habits. Essentially, these frameworks are weaving some uncertainty method into an optimization problem formulation.

Our approach follows the same idea. We will employ the cloud-based uncertainty modeling to formulate an optimization problem, and we will develop some methods to solve the optimization problem. Previously we already mentioned the difficulties of design optimization. Apart from them, we now face additionally a bilevel structure imposed by the uncertainties. One of the many problems coming with a bilevel formulation is the evaluation of the objective function of the outer level. The outer level objective function is actually the solution of the inner level problem, hence an optimization problem itself. Therefore the objective function evaluation can be very expensive.

In view of all the difficulties involved in robust design optimization we will be limited to the development of heuristic optimization methods. The **basic concept** of our approach can be summarized in three fundamental steps within an iterative framework. First, an expert provides the underlying system model, given as a black box model, and all initially available uncertainty information on the input variables of the model. Second, the information is processed to generate a potential cloud. For a given confidence level, the clouds provide confidence regions of relevant scenarios affecting the worst-case of a design. The confidence regions can be formulated as safety constraints for the optimization. Third, optimization methods minimize a certain objective function (e.g., cost, mass) subject to the functional constraints which are represented by the system model, and subject to the safety constraints from the cloud. To this end we develop appropriate heuristic optimization techniques. The results of the optimization are returned to the expert, who will be given an interactive possibility to provide iteratively additional information triggered by the analysis of the results, and to rerun the procedure, on this way adaptively improving the uncertainty model.

1.3 Thesis structure

The content of this dissertation is organized as follows. We start in the next chapter with an overview of different concepts from the field of uncertainty modeling, ranging from the basic fundamentals of probability theory to state-of-the-art methodologies for handling incomplete information. Afterwards in Chapter 3 we will go into the details of our preferred uncertainty model which is based on the clouds formalism. We will figure out how they can be used to cope with a lack of information, and how the concept of potential function based clouds helps in higher dimensional cases. Chapter 4 formulates a general design optimization problem. The related difficulties will be pointed out, and we will present a solution approach to overcome these difficulties. Since the focus of our studies is not on the mere design optimization problem we will formulate and work towards a solution of the associated robust design optimization problem in Chapter 5.

Chapter 2, Chapter 3 and Chapter 4 are self-contained and readable independently of the other chapters. Chapter 5 weaves the methodology developed in Chapter 3 into the solution approach given in Chapter 4.

Chapter 6 presents the functionality of our software implementation of the developed methods. In Chapter 7 we investigate some case studies in which we apply our new methods to real-life robust design optimization problems. The less relevant specification details of the case studies are given in Appendix A and Appendix B.

Finally, the reader will find a summary of all relevant notations, symbols, and abbreviations at the end of the dissertation.

The major parts of the dissertation have been published. The concept of potential clouds is presented in Fuchs & Neumaier [60]. Robust design optimization by means of clouds is studied in Fuchs & Neumaier [59]. The case studies stated in Chapter 7 can be found in Fuchs et al. [58] and Neumaier et al. [118].

Chapter 2

An introductory survey of uncertainty modeling

This chapter presents a survey of conventional and modern approaches to uncertainty handling. For each method, the notation, the type of input information, and the basic concepts will be introduced. We will discuss the necessary assumptions, the rigor of results, and the sensitivity of the results to a lack of information. Typically, the more general a method is, the more expensive it becomes computationally, so we will also comment on computational effort, especially in higher dimensions. Eventually, we will highlight relationships between the presented methods and applications in design optimization.

We start with a section on the fundamentals of probability theory and basic principles used in uncertainty handling. Then we present the several different approaches to uncertainty handling: reliability methods, p-boxes, Bayesian methods, Dempster-Shafer theory, fuzzy sets, convex methods, potential clouds, imprecise probabilities, and a list of other methods.

2.1 Basic principles

Throughout this study we assume familiarity with the basic principles of probability theory. In this section we introduce the notation and fundamental concepts which are the basis for classical methods of uncertainty modeling and of many modern methods as well.

2.1.1 Probability spaces

The traditional approach to define a **probability measure**, pioneered by Kolmogorov [81], is based on a sample space Ω, a σ-algebra $\mathcal{A} \subseteq 2^{\Omega}$ on Ω, and a probability measure μ with $\mu(\Omega) = 1$ defining the associated probability space $(\Omega, \mathcal{A}, \mu)$.

A **random vector** of size n (for $n = 1$, a **random variable**) is defined as a measurable function $\varepsilon : \Omega \to \mathbb{R}^n$, where $\mathbb{R}^n$ is equipped with the Borel σ-algebra $\mathcal{B}$. Any $\varepsilon(\omega) \in \mathbb{R}^n$, $\omega \in \Omega$, is called a **realization** of ε. If not stated otherwise we denote random variables by X and random vectors by $\varepsilon = (\varepsilon^1, \dots, \varepsilon^n)^T$. Let $f : \varepsilon(\Omega) \to \mathbb{R}^n$ be a function such that the composition $f \circ \varepsilon$ is measurable, then $f \circ \varepsilon$ is a random vector which we will write as $f(\varepsilon)$. The **expectation** of a random vector ε is given by

$$\langle \varepsilon \rangle := \int \varepsilon d\mu, \tag{2.1}$$

if ε is integrable (and undefined otherwise). For random vectors $\varepsilon, \varepsilon_\ell$,

$$\begin{aligned}
&\langle a\varepsilon \rangle = a\langle \varepsilon \rangle \text{ for } a \in \mathbb{R}, && (2.2)\\
&\langle \varepsilon_1 + \varepsilon_2 \rangle = \langle \varepsilon_1 \rangle + \langle \varepsilon_2 \rangle, && (2.3)\\
&\langle 1 \rangle = 1, && (2.4)\\
&\langle \varepsilon \rangle \geq 0 \text{ if } \varepsilon \geq 0, && (2.5)\\
&\langle \varepsilon^T \varepsilon \rangle = 0 \Rightarrow \varepsilon = 0, && (2.6)\\
&\langle \varepsilon_n \rangle \to \langle \varepsilon \rangle, \text{ if } \varepsilon_n \to \varepsilon \text{ monotonically.} && (2.7)
\end{aligned}$$

Using these properties axiomatically, the measure theoretic context can be completely hidden. In Whittle [165] it is shown that the probability measure and its relation to expectation via (2.1) can be reconstructed from this axiomatic basis.

A **statement** is a mapping st : $\Omega \to \{\text{true}, \text{false}\}$ such that the characteristic function 1_{st} with

$$1_{\mathrm{st}}(\omega) := \begin{cases} 1, & \text{if } \mathrm{st}(\omega) = \text{true}, \\ 0, & \text{if } \mathrm{st}(\omega) = \text{false} \end{cases} \tag{2.8}$$

is a random variable. If ε is an n-dimensional random vector and $A \subseteq \mathbb{R}^n$ we write loosely $\varepsilon \in A$ for the statement st with $\mathrm{st}(\omega) =$ "$\varepsilon(\omega) \in A$".

We denote the **probability** of the statement st by

$$\Pr(\mathrm{st}) := \langle 1_{\mathrm{st}} \rangle. \tag{2.9}$$

Moreover, we denote the probability of the statement $\varepsilon \in A$, $A \subseteq \mathbb{R}^n$, by

$$\Pr(A) := \Pr(\varepsilon \in A) = \langle 1_{\varepsilon \in A} \rangle = \langle 1_A(\varepsilon) \rangle, \tag{2.10}$$

where 1_A is the conventional characteristic function of the set A,

$$1_A(\varepsilon) := \begin{cases} 1, & \text{if } \varepsilon \in A, \\ 0, & \text{otherwise.} \end{cases} \tag{2.11}$$

Note that

$$\Pr(A \cup B) = \Pr(A) + \Pr(B) - \Pr(A \cap B) \tag{2.12}$$

for two sets $A, B \subseteq \mathbb{R}^n$.

The probability measure of a random variable X is the measure μ_X on $\mathbb{R}$ with

$$\int f(x) d\mu_X(x) = \langle f(X) \rangle \tag{2.13}$$

for all bounded measurable $f : X(\Omega) \to \mathbb{R}$. A random variable X is called **discrete** if it takes discrete values in $\mathbb{R}$. In this case

$$\int f(x) d\mu_X(x) = \sum_{x_i \in X(\Omega)} p_i f(x_i), \tag{2.14}$$

where $p_i = \Pr(X = x_i)$. The random variable X is called **continuous** if it has a **probability density function** (PDF), a continuous function $\rho : \mathbb{R} \to \mathbb{R}^+$

such that $d\mu_X = \rho dx$, where dx is the Lebesgue measure on $\mathbb{R}$.

Table 2.1: Some important probability distributions

Notation	**Description**	**PDF**
$U(a,b)$	uniform distribution in $[a,b]$	$\rho(x) = \begin{cases} \frac{1}{b-a} & \text{if } a \leq x \leq b, \\ 0 & \text{otherwise} \end{cases}$
$N(\mu,\sigma)$	normal distribution with mean μ and variance σ^2	$\rho(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$
$N(\mu,C)$	multivariate normal distribution with mean $\mu \in \mathbb{R}^n$ and covariance matrix $C \in \mathbb{R}^{n\times n}$	$\rho(x) = \frac{1}{\sqrt{(2\pi)^n \cdot \lvert C \rvert}} e^{-\frac{1}{2}(x-\mu)^T C^{-1}(x-\mu)}$
$L(\mu,\sigma)$	lognormal distribution, distribution parameters μ and σ (mean and standard deviation of the associated normal distribution)	$\rho(x) = \frac{1}{\sigma\sqrt{2\pi}} \frac{e^{-\frac{(ln(x)-\mu)^2}{2\sigma^2}}}{z}$ for $x > 0$
$\text{Exp}(\mu)$	exponential distribution with mean μ and variance μ^2	$\rho(x) = \begin{cases} \frac{1}{\mu} e^{-\frac{x}{\mu}} & \text{if } x \geq 0, \\ 0 & \text{if } x < 0 \end{cases}$
$\Gamma(\alpha,\beta)$	gamma distribution with mean $\alpha\beta$ and variance $\alpha\beta^2$	$\rho(x) = \frac{1}{\beta^\alpha \Gamma(\alpha)} x^{\alpha-1} e^{-\frac{x}{\beta}}$ for $x > 0$
$\chi^2(n)$	chi^2 distribution with n degrees of freedom	corresponds to a $\Gamma(\frac{n}{2}, 2)$ distribution
$\text{Cauchy}(m,\Theta)$	Cauchy distribution with median m and parameter $\Theta > 0$	$\rho(x) = \frac{\Theta}{\pi(\Theta^2 + (x-m)^2)}$

A **cumulative distribution function** (CDF) of a random variable X is a monotone function $F : \mathbb{R} \to [0,1]$ with

$$F(t) = \Pr(X \leq t), \tag{2.15}$$

$$\lim_{t\to\infty} F(t) = 1, \tag{2.16}$$

$$\lim_{t\to-\infty} F(t) = 0. \tag{2.17}$$

Table 2.1 shows some examples of continuous CDFs. The relation between the

PDF ρ and the CDF F of a random variable X is given by

$$F(t) = \int_{-\infty}^{t} \rho(x)dx. \tag{2.18}$$

Note that $F(X)$ as a random variable is $U(0,1)$ distributed.

We denote, e.g., $F_{U(0,1)}$ as the CDF of a $U(0,1)$ distributed random variable, or more generally $F_{\text{distribution}}$ the CDF of a random variable with the indexed distribution.

The use of CDFs and PDFs extends to higher dimensions pointwise via

$$F(t^1, \dots, t^n) = \Pr(X^1 \leq t^1, \dots, X^n \leq t^n), \tag{2.19}$$

$F : \mathbb{R}^n \to [0,1]$, and

$$F(t) = \int_{-\infty}^{t^1} \int_{-\infty}^{t^2} \dots \int_{-\infty}^{t^n} \rho(x^1, x^2, \dots, x^n)dx^n \dots dx^1, \tag{2.20}$$

$\rho : \mathbb{R}^n \to [0,1]$. F is called the **joint CDF**, ρ is called **joint PDF**.

2.1.2 Reliability and failure

To employ probability theory in design safety problems, we need to define failure probabilities p_f and reliability R. The **failure probability** of a fixed design is the probability that the random vector ε lies in a set $\mathbb{F}$ of scenarios which lead to design failure. The **reliability** is the probability that the design will perform satisfactorily, i.e.,

$$R := \Pr(\varepsilon \notin \mathbb{F}) = 1 - \Pr(\varepsilon \in \mathbb{F}) = 1 - p_f, \tag{2.21}$$

so determining R and p_f are equivalent problems. A third important notion is that of a confidence region for ε. A set C_α is a **confidence region** for the confidence level α if

$$\Pr(\varepsilon \in C_\alpha) = \alpha. \tag{2.22}$$

The relation between confidence regions and failure probabilities can be seen as follows. Assume that we have determined a confidence region C_α for the random vector ε, and C_α does not contain a scenario which leads to design failure, i.e., $C_\alpha \cap \mathbb{F} = \emptyset$. Then $\Pr(C_\alpha \cup \mathbb{F}) = \Pr(C_\alpha) + \Pr(\mathbb{F}) \leq 1$. Hence $p_f = \Pr(\mathbb{F}) \leq 1 - \Pr(C_\alpha) = 1 - \alpha$, the failure probability is at most $1 - \alpha$. For the reliability $R = 1 - p_f$ we get $R \geq \alpha$.

2.1.3 Incomplete Information

To make use of probabilistic concepts one often assumes that for the random vector ε involved the joint distribution F is precisely known, provided by an objective source of information. In many design problems, the sources of information are merely subjective, provided by expert knowledge. Additionally, in higher dimensions joint distributions are rarely available, and the typically available distribution information consists of certain marginal distributions. A 1-dimensional **marginal distribution** of the component ε^i of the random vector ε is given by

$$F_i(t) := F(c^1, \ldots, c^n), \text{ where } c^k = \infty \text{ for } k \neq i \text{ and } c^k = t \text{ for } k = i. \quad (2.23)$$

Often one simply fixes the CDF as normally distributed, arguing with the central limit theorem: a sufficiently large amount of statistical sample data justifies the normal distribution assumption. The critical question is, what is sufficiently large in higher dimensions? The generalized Chebyshev inequality (2.24) gives rigorous bounds for the failure probability $p_f = \Pr(\mathbb{F})$, in case that $\mathbb{F}(r) = \{\varepsilon \mid ||\varepsilon||_2 \geq r\}$, r a constant radius. If the components of $\varepsilon = (\varepsilon^1, \ldots, \varepsilon^n)$ are uncorrelated, have mean 0 and variance 1, we get from Neumaier [117]

$$p_f = \Pr(\mathbb{F}) \leq \min(1, \frac{n}{r^2}), \quad (2.24)$$

and this bound can be attained.

The failure probability bounds from (2.24) differ significantly from those of a normal distribution as shown in Figure 2.1. If we assume a multivariate normal distribution for ε, uncorrelated is equivalent to independent. The bounds for normal distribution assumption can then be computed from the $\chi^2(n)$ dis-

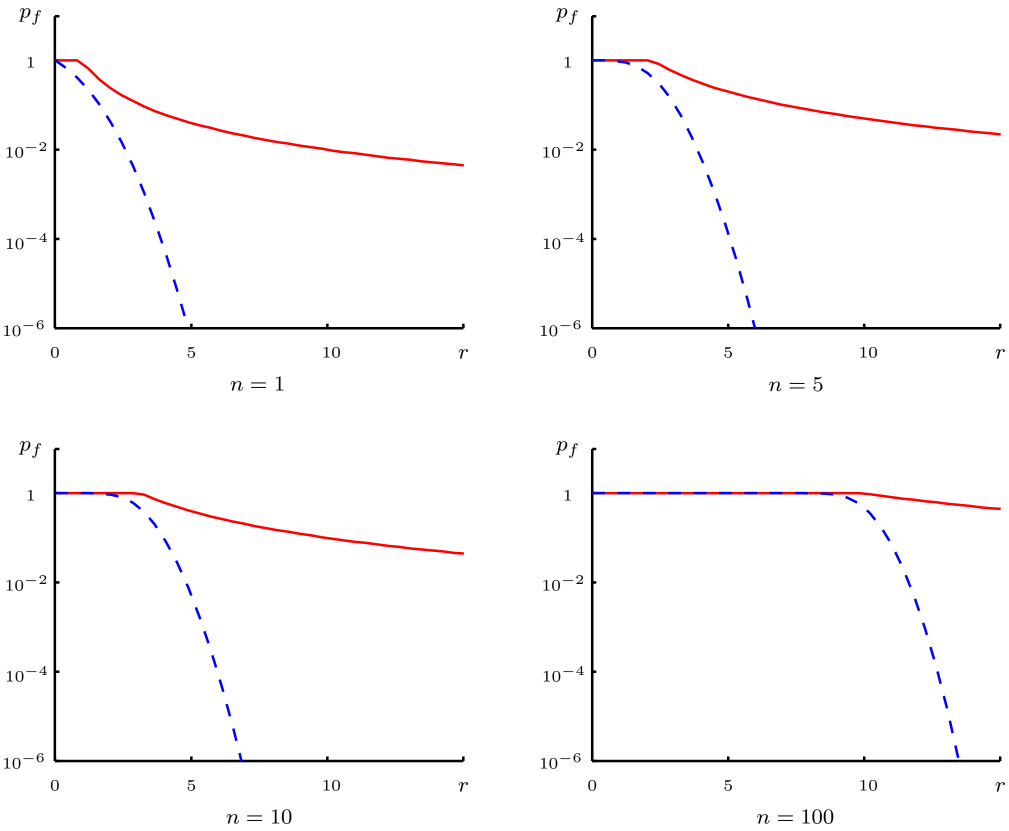


Figure 2.1: Failure probability p_f for the failure set $\mathbb{F}(r)$ in different dimensions n, bounded from above by the Chebyshev inequality (solid line) and computed from a normal distribution (dashed line), respectively.

tribution (a special case of the gamma distribution: $\chi^2(n) = \Gamma(\frac{n}{2}, 2)$): the sum of squares of independent standard normally distributed random variables $\varepsilon^1, \dots, \varepsilon^n$ is $\chi^2(n)$ distributed, so $p_f = 1 - F_{\chi^2(n)}(r^2)$. We see that the normal distribution assumption can be much too optimistic compared with the optimal worst-case bounds from (2.24).

An alternative justification of the normal distribution assumption is the **maximum entropy principle**, if the available information consists of mean and standard deviation only. The principle of maximum entropy originates from information theory, cf. Shannon & Weaver [145], and is utilized in many fields of applications, cf., e.g., Grandy & Schick [62]. Entropy of a random variable X with PDF ρ can be defined by

$$H(X) := -\int_{-\infty}^{\infty} \rho(x)\log(\rho(x))dx. \tag{2.25}$$

The intuitive meaning of entropy is: the larger the entropy the less information (relative to the uniformly distributed improper prior) is reflected in the probability measure with density ρ. In order to define a probability measure given incomplete information, the principle of maximum entropy consists in maximizing the entropy subject to constraints imposed by the information available. For example, in the case of given mean and standard deviation this ansatz leads to a normal distribution, in case of given interval information it leads to a uniform distribution assumption. Note that as soon as we employ the maximum entropy distribution as a probability measure we pretend to have more information than actually available. Hence critical underestimation of failure probabilities may show up.

In a nutshell, the concept of random variables and probability spaces enables one to derive rigorous statements about failure probabilities and reliability. But they require the probability measure to be precisely known. Otherwise, tails of CDFs can be critically underestimated, so the estimation of failure probabilities becomes quite poor. In Elishakoff [39] one can find a demonstration that straightforward probabilistic computations are highly sensitive to imprecise information. Imagine a CDF is known almost precisely, but with a small deviation in some distribution parameter. This may easily lead to a situation as shown in Figure 2.2, which illustrates the fact that the failure probability is often underestimated (here by the factor 2).

In the univariate case it is simple to overcome problems with lack of information. One can apply Kolmogorov-Smirnov (KS) statistics as a powerful tool. Assume that the uncertainty information is given by empirical data on a random variable X, e.g., a small set of sample points $x_1, \dots, x_N$. The **empirical distribution** $\widetilde{F}$ is defined by

$$\widetilde{F}(\xi) := \sum_{\{j \mid x_j \leq \xi\}} \frac{1}{N}. \tag{2.26}$$

The KS test uses $D := \max |\widetilde{F} - F|$, the maximum deviation of $\widetilde{F}$ from F, as its test statistics, and it can be shown that $\sqrt{N}D$ converges in distribution to

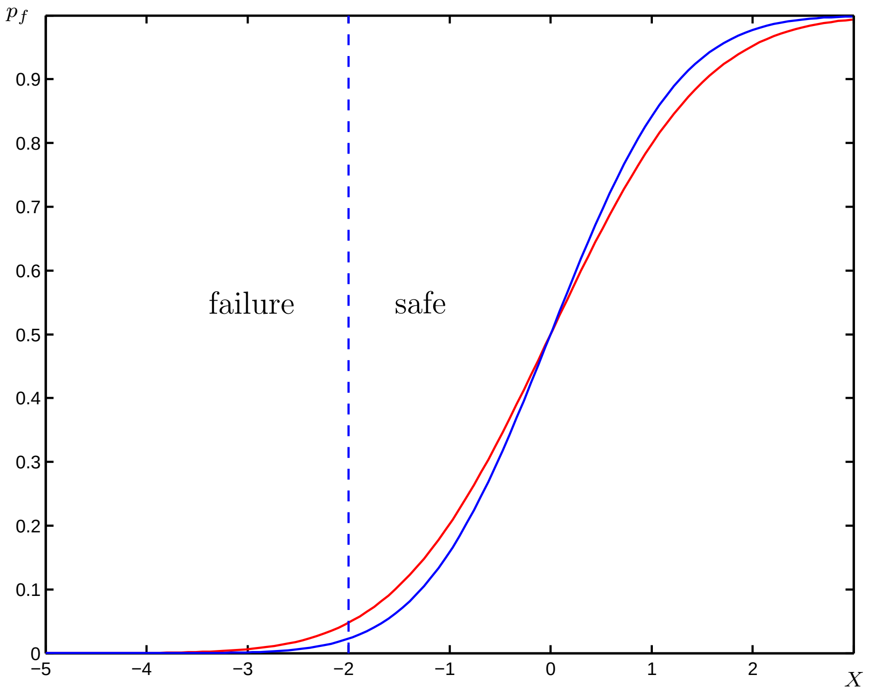


Figure 2.2: Small deviation in a distribution parameter (here 20% difference in the standard deviation of two normal distributions) can lead to critical underestimation of p_f for a random variable X. Here p_f is underestimated by the factor 2, if the design failure set was $\mathbb{F} = \{X \mid X \leq -2\}$.

the Kolmogorov function

$$\phi(\lambda) := \sum_{k=-\infty}^{+\infty} (-1)^k e^{-2k^2\lambda^2} \tag{2.27}$$

for $N \to \infty$, cf. Kolmogorov [82]. Conversely, if we choose a fixed confidence level α, we can compute D from

$$D = \frac{\phi^{-1}(\alpha)}{\sqrt{N} + 0.12 + \frac{0.11}{\sqrt{N}}}, \tag{2.28}$$

cf. Press et al. [131], and thus find a maximum deviation of the unknown F from the known $\widetilde{F}$. That means that we have non-parametric bounds $[\widetilde{F} - D, \widetilde{F} + D]$ enclosing F with confidence α, only given the knowledge of $x_1, \dots, x_N$.

In case of high-dimensional random vectors, classical probability theory has no means to cope with scarce data as in the univariate case with KS bounds. Although multivariate CDFs can be defined as in the 1D case using the componentwise partial order in $\mathbb{R}^n$, the computational effort for obtaining higher dimensional PDFs and their numerical integration prohibit the reliable use of standard probabilistic methods in higher dimensions.

2.1.4 Safety margins

A simple and widely spread non-probabilistic uncertainty model is based on so-called **safety margins**. This model is applied when very little information is available, in situations where most information is provided as interval information.

There are different kinds of sources for interval information, e.g., measurement accuracy. Safety margins are a special kind of interval information, namely one which is provided subjectively by an expert designer, as typically the case in early design phases. If additional information is available, like marginal distributions or safety margins from further experts, the safety margins approach cannot handle it and thus loses some valuable information. Since safety margins are highly subjective information one cannot expect rigorous results for the safety analysis. However, engineers hope to achieve reasonably conservative bounds by using conservative margins.

The first approach – a tool to handle all kinds of interval information – is **interval analysis**, cf. [107], [115]. We write $X \in [a, b]$ for $a \leq X \leq b$ in the univariate case; in the higher dimensional case $\varepsilon = (\varepsilon^1, \dots, \varepsilon^n)$, we define interval information $\varepsilon \in [a, b]$ pointwise via $a^1 \leq \varepsilon^1 \leq b^1, \dots, a^n \leq \varepsilon^n \leq b^n$, and call $[a, b]$ a **box**. We always interpret equalities and inequalities of vectors componentwise. In the following we present two frequent approaches to handle the incoming interval information.

Assume that the cost or gain (or another assessment) function $s : \mathbb{M} \subseteq \mathbb{R}^n \to \mathbb{R}^m$, with design space $\mathbb{M}$, models the response function of the design, and the information about the uncertain input vector ε is given by the bounds

$\varepsilon \in [a,b] \subseteq \mathbb{M}$. By way of interval calculations one achieves bounds componentwise on $s(\varepsilon)$ – also called an **interval extension** of s.

Computing an interval extension is often affected by overestimation. A variable $\varepsilon^i \in [a^i, b^i]$ should take the same value from the interval $[a^i, b^i]$ each time it occurs in an expression in the computation of s. However, this is not considered by straightforward interval calculations, so the range is computed as if each time the variable ε^i occurs it can take a different value from $[a^i, b^i]$, leading to an enclosure which may be much wider than the range for $f(\varepsilon)$. One possible way out is based on Taylor series, cf. Makino & Berz [95]. Nonlinear interval computations in higher dimensions may become expensive, growing exponentially with n, but can often be done efficiently and complemented by simulation techniques or sensitivity analysis, as we will see later. Note that in case that s is given as a black box evaluation routine – as in many real-life applications – the interval extension cannot be determined rigorously anyway. Also interval methods are often not applicable as a toolbox, but require problem specific expert knowledge to overcome overestimation issues.

In the literature we find much utilization of interval computations in uncertainty modeling. Analyzing the statistics for interval valued samples one seeks to bound mean or variance, which are then also interval valued, cf. Kreinovich [87]: Finding an upper bound on the variance is NP-hard, a lower bound can be found in quadratic time. The field of applications of interval uncertainty for uncertainty handling is vast, e.g., [47], [91], [111], [112], [129].

Also probability theory proper makes use of non-probabilistic interval uncertainty models. For example, consider a Markov chain model with transition matrix (p_{ij}), where the transition probabilities p_{ij} are uncertain, and only given as intervals. Then one can build a generalized Markov chain model, cf. [148], [149]. In [55] one can find a study of imprecise transition probabilities in Markov decision processes.

The second approach to handle safety margin interval information is a simplification of the information by **fixing each uncertain variable** $\varepsilon^i \in [a^i, b^i]$ at the value of one of the safety margins a^i or b^i, and simply insert this value, for instance a^i for all i, as worst-case scenario to compute the worst-case design response $s(a)$. The decision where to fix the worst-case scenario is taken merely subjectively or via a list of relevant cases. A designer may overestimate intentionally the subjective safety margin assignments, e.g., by adding $20\% = 2 \cdot 0.1$ to the nominal interval bounds for a variable, i.e., $\varepsilon \in [a - 0.1(b-a), b + 0.1(b-a)]$, in order to be suitably conservative in computing the worst-case design response.

The computational effort with this latter approach is not very high and also applies well in higher dimensions. Actually, there is no extra effort in addition to the cost for evaluating s involved in this simple uncertainty model.

A field where safety margins are very popular is MDO. In many cases, in particular in early design phases, it is common engineering practice to combine the assignment of safety margins with an iterative process of refining or coarsening the margins, while converging to a robust optimal design. The refinement of the intervals is done by experts who assess whether the worst-case scenario determined for the design at the current stage of the iteration process is too pessimistic or too optimistic. The goal of the whole iteration includes both optimization of the design and safeguarding against uncertainties. The achieved design is thus supposed to be robust. This procedure enables a very dynamic design process and quick interaction between the involved disciplines.

All in all, safety margins allow for a simple, efficient handling of uncertainties, also in large-scale problems, if no information is available but an interval bounding from a single source. Otherwise, we have to look for improved methods, which can handle more uncertainty information. It should be remarked that in most cases, even in early design stages, there is more information available than assumed for the safety margin approach.

2.1.5 Safety factors

Remember the concept of failure probabilities as introduced in Section 2.1.2. The failure probability was defined as $p_f = \Pr(\mathbb{F})$, where $\mathbb{F}$ was the set of events which lead to design failure. Let $s : \mathbb{R}^n \to \mathbb{R}$ be the design response of a fixed design for uncertain inputs $\varepsilon \in \mathbb{R}^n$. Assume that there is a limit state ℓ for which $s(\varepsilon) < \ell$ means design failure, and $s(\varepsilon) \geq \ell$ represents satisfying design performance, i.e., that $\mathbb{F}$ is defined by

$$\mathbb{F} = \{\varepsilon \mid s(\varepsilon) < \ell\}. \tag{2.29}$$

The idea behind safety factors is to build the design in a way that the expected value of $s(\varepsilon)$ is greater than the limit state $\ell > 0$ multiplied by a factor $k_{\text{safety}} > 1$, called the **safety factor**. In other words, a design should fulfill

$$\langle s(\varepsilon)\rangle \geq k_{\text{safety}}\ell. \tag{2.30}$$

where the expectation has to be suitably estimated. For $s(\varepsilon) \in \mathbb{R}^m$ we interpret this definition componentwise for each safety requirement on the design responses $s^1, \dots, s^m$, and $\ell = (\ell^1, \dots, \ell^m)$.

Conversely, suppose that we are given a fixed design and $\langle s(\varepsilon)\rangle \geq \ell$, $s(\varepsilon) \in \mathbb{R}$, $s \in C^1$. Define the maximal feasible safety factor as $k := \frac{\langle s(\varepsilon)\rangle}{\ell}$. To see the relation between k and the design failure probability p_f we assume that we have determined p_f for the fixed design, then $p_f = \Pr(s(\varepsilon) \leq \ell) = F_s(\ell)$. Here F_s is the CDF of $s(\varepsilon)$ with density ρ_s given by

$$\rho_s(x) = \rho(s^{-1}(x)) \cdot |\det s'(x)|^{-1}, \tag{2.31}$$

with $|\det s'(x)|$ the absolute value of the determinant of the Jacobian of s. Hence we get $\ell = F_s^{-1}(p_f)$, assuming that F_s is invertible, and

$$k = \frac{\langle s(\varepsilon)\rangle}{F_s^{-1}(p_f)}. \tag{2.32}$$

As we are applying standard methods from probability theory to compute safety factors, precise knowledge of ρ and of the **limit state function** $s(\varepsilon) - \ell$ is required to achieve rigorous probability statements.

In the lower dimensional case, if ρ is unknown, but a narrow bounding interval and certain expectations (e.g., means and covariances) for the random vector ε are known, safety factors can still be well described approximately. The expectation of smooth functions s of ε is then achievable from the Taylor series for s, cf., e.g., Berleant et al. [11], since expectation is a linear operator on random variables – similar to Taylor models for interval computations. The problems mentioned in Section 2.1.3 concerning lack of information in the higher dimensional case remain.

Probabilistic computation of safety factors is not as much affected by subjective opinions of the designer as, for example, safety margins. Safety factors are directly associated with required reliability. One important subjective decision is how to fix the required reliability or the admissible failure probability, respectively. The decision can be based, e.g., on the assessment of failure cost or on regulations by legislation.

2.1.6 Simulation methods

Simulation methods are ubiquitous tools, and uncertainty handling is one of their application fields. Simulation means computational generation of sample points as realizations of a random vector ε, assuming that the joint CDF, marginal CDFs, or interval bounds are given. Thus the involved not necessarily probabilistic uncertainties involved are simulated, which gives rise to the term simulation methods. Simulation methods are also referred to via the terms **random sampling** or **Monte Carlo sampling**.

After sample generation, the design response $s : \mathbb{M} \to \mathbb{R}^m$ is evaluated for each generated sample point. If all or at least a reasonable majority of the points meet the safety requirements $s(\varepsilon) \geq \ell$, the design is considered to be safe.

The core part of simulation is the sample generation. There is a large number of different techniques addressing it. I will shortly introduce three classical variants, one based on CDF inversion, one based on Markov chains, and Latin hypercube sampling.

The first one is a straightforward consequence of the CDF definition. Assume that we wish to generate a sample for the random variable X given its continuous invertible joint CDF F. Then use the result that $F(X)$ as a random variable is $U(0,1)$ distributed, cf. Section 2.1.1. Hence to generate a sample, first generate uniformly distributed sample points $x_1^u, \dots, x_N^u$ and transform them to the distribution of X by $F^{-1}(x_1^u), \dots, F^{-1}(x_N^u)$. This method apparently requires inversion of the CDF and is particularly not applicable in higher dimensions.

The second variant is based on Markov chains. Let the sequence $(X_k)_{k\geq 1}$ be a Markov chain with transition probabilities $p_{ij} = \Pr(X_{k+1} = j \mid X_k = i)$ for the states i, j. Here, $\Pr(X_{k+1} = j \mid X_k = i)$ denotes the **conditional probability** that a transition to the state j is performed given the state i.

Conditional probability of an event A given the event B is defined as

$$\Pr(A \mid B) := \frac{\Pr(A \cap B)}{\Pr(B)}. \tag{2.33}$$

The Markov chain is called **reversible** if $p_i p_{ij} = p_j p_{ji}$, where p_i is the stationary probability of the state i. The result of interest for Markov chain random sampling is that a reversible Markov chain visits each state i with a relative frequency equal to its associated equilibrium probability p_i. The sampling algorithm constructs a reversible Markov chain using the Metropolis approach, cf. Metropolis et al. [100], or the more general Hastings method, cf. Hastings [66], by way of a rejection method. The rejection rule assures that it is not necessary to compute p_i, but only the easy-to-compute quotient $\frac{p_j}{p_i}$ of two different states which is independent of the dimension of X_n. This makes the method highly attractive in higher dimensions.

Often **importance sampling** is used to speed up simulation techniques by a reduction of the number of required simulations, e.g., [69], [79], [155]. The sample points are generated from a different distribution than the actual distribution of the involved random variables. The sampling density is weighted by an importance density, e.g., a normal distribution with standard deviation σ depending on where the most probable failure points are expected, for instance, depending on the curvature of s. Thus the generated sample is more likely to cover the 'important' regions for the safety analysis.

The third variant is the so-called **Latin hypercube sampling** (LHS), cf. McKay et al. [98]. First, one determines a finite grid of size N^n, where N is the desired number of sample points and n is the dimension of the random vector for which we want to generate a sample. The grid is preferably constructed such that the intervals between adjacent marginal grid points have the same marginal probability. The N sample points $x_1, x_2, \ldots, x_N$, $x_i = (x_i^1, \ldots, x_i^n) \in \mathbb{R}^n$ are then placed to satisfy the Latin hypercube requirement,

$$\text{For } i,k \in \{1,\ldots,N\},\ j \in \{1,\ldots,n\} :\ x_i^j \neq x_k^j \text{ if } k \neq i, \tag{2.34}$$

illustrated in Figure 2.3.

This procedure introduces some preference for a simple structure, i.e., we disregard correlations, tacitly assuming independence. The advantage of the method is that the full range of ε is much better covered than with a Markov

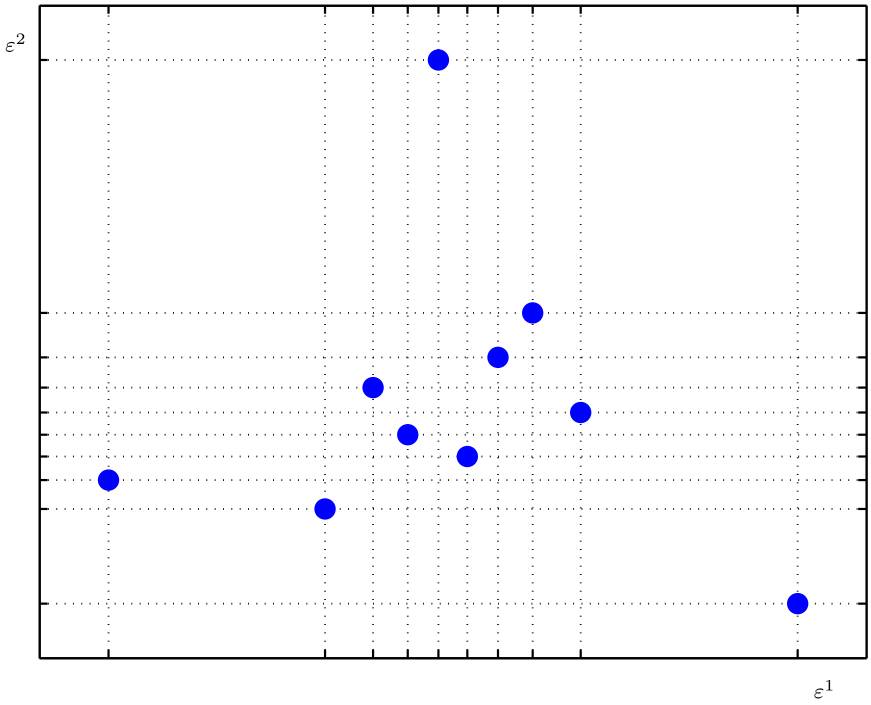


Figure 2.3: LHS of 10 sample points in two dimensions for a marginal (symmetric) normal distribution of ε^1 and a marginal (asymmetric) Γ distribution of ε^2. The intervals between adjacent marginal grid points have the same marginal probability. The condition (2.34) requires exactly one sample point in each row and column of the grid.

chain based method, giving deeper insight to the distribution tails of ε. Hence failure probabilities can be better estimated. Moreover, one does not require more sample points for higher n, so the application of LHS in higher dimensions is still attractive.

Considering the rigor of the results one should be aware of the fact that no estimation of failure probabilities computed from a simulation technique is a rigorous bound. These methods are based on the law of large numbers, and their results are only valid for a sufficient amount of sample points. It is difficult to assess what 'sufficient amount' means in a higher dimensional space; one might need to generate an excessively large number of sample points for estimating very small failure probabilities. That is why simulation methods are endangered to critically underestimate CDF tails, cf. Ferson [44]. It gets particularly dangerous when the CDFs to sample from are unknown.

On the other hand, simulation methods are computationally very efficient, they can be parallelized [89], and also apply well in higher dimensions, where almost no alternatives exist at present.

Another important aspect comes with **black box response functions** s. They principally impose no additional difficulties applying simulation methods. However, if the computational cost for evaluating s is very high, problems will arise as simulation typically requires many evaluations, hence is limited to simple models for s, often surrogate functions (cf., e.g., [37], [72]) for more complex models.

As mentioned earlier simulation techniques have many applications, e.g., the computation of multi-dimensional integrals. They are related to many uncertainty methods, also non-probabilistic ones like interval uncertainty.

2.1.7 Sensitivity Analysis

Sensitivity analysis is actually not an independent uncertainty method itself, it rather applies in several different fields one of which is uncertainty handling. Sensitivity analysis investigates the variability of a model function output $f(\varepsilon)$ $f : \mathbb{R}^n \to \mathbb{R}$, $\varepsilon = (\varepsilon^1, \varepsilon^2, \dots, \varepsilon^n)^T$, with respect to changes in the input variables ε^i.

To this end one can follow different approaches, e.g., investigate the partial derivatives of f if they are available, using $\frac{\partial f}{\partial \varepsilon^i}$ as an indicator for the influence of ε^i on f. One can also vary a subset of all single input variables ε^i of f while keeping all other inputs constant. Then one assesses the impact of this subset by the variability of the output f by means of some uncertainty methods introduced in this chapter, e.g., fuzzy set or simulation methods. Thus one hopes to achieve a dimensionality reduction of f fixing those input variables which turn out to have little influence on f. Frequently, e.g., Kreinovich et al. [88], one assumes monotonicity of f and interval uncertainty of ε, since this enables the use of very fast techniques in higher dimensions, the effort is growing only linearly in the dimension n.

As a particular case of handling interval uncertainties in high dimensions with computationally expensive black box response functions s we mention the Cauchy distribution based simulation for interval uncertainty, cf. KREINOVICH & FERSON [89]: Assuming that the intervals are reasonably small, e.g., given as measurement errors, it is reasonable to assume that s is linear. Generate N independent sample points for the measurement errors from the scaled Cauchy$(0, 1)$ distribution, which is easy as the inverse CDF $F^{-1}_{\text{Cauchy}(0,1)}$ is known explicitly in this case. Linear functions of Cauchy distributed variables are again Cauchy distributed, cf. WEISSTEIN [164], with an unknown parameter Θ. Having estimated the parameter Θ, e.g., by means of a maximum likelihood estimator, one can infer probabilistic statements about errors in s which are Cauchy$(0, \Theta)$ distributed. Thus this method exploits the characteristics of a Cauchy distribution to produce results the accuracy of which can be investigated statistically depending on N, also for low N in case of expensive s. No derivatives are required, only N black box evaluations of s.

Applications of sensitivity analysis can be found, e.g., in [125], [130].

2.1.8 Transformations to standard normal space

While sampling methods, introduced in the last section, may use transformations to generate a sample with a desired distribution from a set of a uniformly distributed sample points, some uncertainty methods instead transform the probability space to the so-called **standard normal space**. The transformation $\mathcal{T} : \mathbb{R}^n \to \mathbb{R}^n$ of ε to standard normal space is sought to be constructed such that $u = \mathcal{T}(\varepsilon)$ is standard normally distributed (i.e., with the identity as covariance matrix and mean zero). The transformation $\mathcal{T}$ is not unique. Two ways of constructing computationally are the Rosenblatt transformation [137] and the Nataf transformation [113].

The **Rosenblatt transformation** is given by successive conditioning. A conditional distribution $F_i(t^i \mid t^1, t^2, \dots, t^{i-1})$ is the CDF of ε^i conditional on $\varepsilon^1 = t^1, \varepsilon^2 = t^2, \dots, \varepsilon^{i-1} = t^{i-1}$. We define

$$\begin{aligned}\mathcal{T}(\varepsilon) := (\Phi^{-1}(F_1(\varepsilon^1)), \Phi^{-1}(F_2(\varepsilon^2 \mid \varepsilon^1)), \Phi^{-1}(F_3(\varepsilon^3 \mid \varepsilon^1, \varepsilon^2)), \dots,\\ \Phi^{-1}(F_n(\varepsilon^n \mid \varepsilon^1, \varepsilon^2, \dots, \varepsilon^{n-1}))), \qquad (2.35)\end{aligned}$$

where Φ is the univariate standard normal distribution. Unfortunately this transformation requires the complete knowledge of the joint or conditional CDFs of ε. But in practice there is rarely more information available than marginal distributions or other incomplete information. Thus the Rosenblatt transformation mainly serves as a theoretical tool.

The **Nataf transformation** takes random variables with known marginal distributions and correlations as input. The marginal distributions are joined via the concept of copulas to achieve a standard normal joint CDF.

The first transformation step is transformation of each variable $\varepsilon^1, \ldots, \varepsilon^n$ to a uniform distribution via the marginal CDF F_i. The second step is a **copula**, cf. Sklar [147]. A copula is a multivariate CDF with uniformly distributed 1-dimensional marginal CDFs. For the Nataf transformation the copula is a multivariate $N(\mu, C)$ distribution. To this end one first transforms the marginal CDFs to marginal $N(0, \sigma_i)$ distributions Φ_i, where σ_i is the given standard deviation of ε^i. We have

$$\mathcal{T}(\varepsilon) := (\Phi_1^{-1}(F_1(\varepsilon^1)), \ldots, \Phi_n^{-1}(F_n(\varepsilon^n))). \tag{2.36}$$

It remains to estimate the covariance matrix C of the CDF of $u = \mathcal{T}(\varepsilon)$. The covariance matrix has to be determined by suitably transforming the input correlations, e.g., cf. [121]. If the input information is based on statistical sample data one can directly estimate C from the transformed data u.

Finally, a principal axis transformation completes the transformation to a standard normal distribution. Further ways of finding $\mathcal{T}$ are studied in [68], [121].

2.2 Reliability methods

Reliability methods are a very popular approach based on the concepts of reliability and failure probability and transformation to standard normal space, cf. Rackwitz [134]. They represent a significant improvement in computational modeling of reliability compared to rather old-fashioned methods like safety factors.

In order to investigate the failure probability p_f given the joint CDF F, or at least marginal CDFs, of the involved random vector ε, one first applies a coordinate transformation $u = \mathcal{T}(\varepsilon)$ to standard normal space, cf. Section 2.1.8. Then the failure surface $\mathbb{F} = \{s(\varepsilon) < \ell\}$ is approximated and embedded in an optimization problem to estimate p_f.

Once a $\mathcal{T}$ is found the new coordinates u live in standard normal space, that means the level sets of the density of u are $\{u \mid ||u||_2 = \text{const}\}$, due to the shape of the multivariate normal distribution. Let $s(u)$ be the design response in the transformed coordinates. Then the most probable failure point u^* from the failure set $\mathbb{F} = \{u \mid s(u) < \ell\}$ is the solution of

$$\begin{aligned} &\min_u \ ||u||_2 \\ &\text{s.t.} \ \ s(u) < \ell \end{aligned} \tag{2.37}$$

i.e., the point from $\mathbb{F}$ with minimal 2-Norm. This critical point is called β-point, and

$$p_f \approx \Phi(-\beta) \tag{2.38}$$

approximates the failure probability, where $\beta = ||u^*||_2$ and Φ denotes the CDF of the univariate $N(0, 1)$ distribution.

Thus we have reduced the estimation of p_f to the standard problem of finding $\mathcal{T}$ and the remaining problem of solving the optimization problem (2.37). The latter is a nonlinear optimization problem with all the problems that come with it. Even if the limit state function is convex, after transformation it may become a strongly non convex problem in case that the CDF F significantly differs from a normal distribution. Using a linear approximation of the limit state function in the computation of β is called **first order reliability method** (FORM), a quadratic approximation is called **second order reliability method** (SORM).

One hopes that a unique solution for β exists; however in general, there is usually no guaranteed global and unique solution for this optimization problem. Another problem about this approach is that the β-point found may not be representative for the failure probability. A discussion on the involved optimization problem can be studied in Der Kiureghian & Dakessian [31], investigating difficulties like multiple β-points. The entailed difficulties require some caveats assessing the results of reliability methods: the methods may fail to estimate p_f correctly without warning the user. Especially when additional problems appear – like higher dimensionality or black box response functions s – the reliability methods become less attractive in many large-scale real-life situations. It should be remarked that the search for u^* can be supported by sampling and simulation techniques like importance sampling, cf. Section 2.1.6, as means for corrections and reduction of the computational effort, e.g., [96].

Reliability methods are associated with design optimization within the field of reliability based design optimization (RBDO). Instead of the often occurring bilevel problem formulation (i.e., design optimization in the outer level, worst-case scenario search in the inner level) one formulates a one level problem as follows, cf. [78], [108]. Let $s_{\mathcal{T}} = s_{\mathcal{T}}(\theta, u) = s(\theta, \mathcal{T}(\varepsilon))$, the design response in transformed coordinates, with the controllable design vector θ which fully specifies the design. Let $g(\theta)$ be the objective function, e.g., the cost of the design or the cost of failure. One seeks to minimize g subject to some reliability constraint $p_f \leq p_a$ where $p_f = \Pr(s(\theta, u) < \ell)$ is approximated by equation (2.38), and p_a the fixed admissible failure probability. We get

$$\begin{aligned} \min_{\theta} \ & g(\theta) \\ \text{s.t.} \ & \Phi(-\beta) \leq p_a \\ & \theta \in T \end{aligned} \tag{2.39}$$

where T is the set of possible design choices. For $s \in \mathbb{R}^m$ we have several constraints $\Phi(-\beta_i) \leq p_a$, $i = 1, \ldots, m$.

Usually simulation techniques are employed to solve (2.39), e.g., [139]. In [126] it is suggested to use Monte Carlo methods to check the probabilistic constraints, and to train a neural network to check the deterministic constraints, or even both probabilistic and deterministic. This can be implemented as parallelized computations which improves computation time significantly. In any case, one should be aware that one uses a soft solution technique on top of a soft uncertainty model.

2.3 p-boxes

A p-box – or p-bound, or probability bound – is an enclosure of the CDF of a univariate random variable X, $F_l \leq F \leq F_u$, in case of partial ignorance about specifications of F. Such an enclosure enables, e.g., to compute lower and upper bounds on expectation values or failure probabilities.

There are different ways to construct a p-box depending on the available information about X. We illustrate it with four examples, cf. Ferson et al. [46], visualized in Figure 2.4.

1. Assume that the shape of F is given (e.g., normal distribution), and bounds on distribution parameters $\mu_l \leq \mu \leq \mu_u$ and $\sigma_l \leq \sigma \leq \sigma_u$. The p-box construction then amounts to finding the extremal possible normal distributions with parameters in the specified ranges, which might be a complex optimization problem. However, due to the shape of the normal distribution one only has to compute 4 distributions for the parameter combinations $(\mu_l, \sigma_l), (\mu_u, \sigma_u), (\mu_l, \sigma_u), (\mu_u, \sigma_l)$, and form the envelope to determine the enclosure of F – similar simple p-box construction mechanisms exist for different convenient distribution shapes like lognormal, uniform, exponential.

2. Assume that we are only given information about bounds $X \in [a, b]$, and mean μ of X, $a < \mu < b$. To determine $F_u(t)$ consider $t \leq \mu$. The maximal CDF at t, that is the maximal probability measure on $\{X \leq t\}$ must satisfy the condition on the average $F_u(t)t + (1 - F_u(t))b = \mu$ to have at least mean μ. We get

$$F_u(t) = \begin{cases} 0 & \text{if } t < a, \\ \min(1, \frac{b-\mu}{b-t}) & \text{if } a \leq t < b, \\ 1 & \text{if } t \geq b. \end{cases} \tag{2.40}$$

For F_l we get analogously

$$F_l(t) = \begin{cases} 0 & \text{if } t < a, \\ \max(0, \frac{t-\mu}{t-a}) & \text{if } a \leq t < b, \\ 1 & \text{if } t \geq b. \end{cases} \tag{2.41}$$

3. Assume that we know a bit more: $X \in [a, b]$, and the median m of X. Then

$$F_l = \frac{1}{2}(1_{[m,\infty]} + 1_{[b,\infty]}), \tag{2.42}$$

$$F_u = \frac{1}{2}(1_{[a,\infty]} + 1_{[m,\infty]}) \tag{2.43}$$

constructs the p-box.

4. Assume that we have empirical data for X. Then we can construct a p-box with KS statistics, cf. (2.28), after fixing a confidence level α.

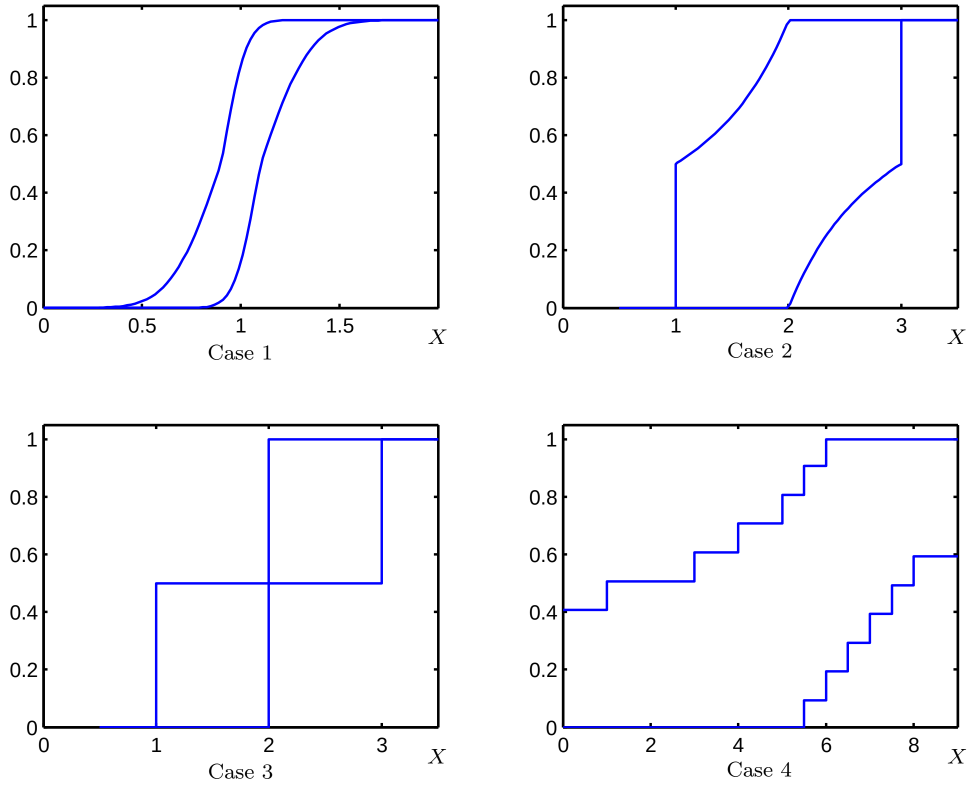


Figure 2.4: Four examples for p-box construction. Case 1 with $\mu_l = 0.9$, $\mu_u = 1.1$, $\sigma_l = 0.1$, $\sigma_u = 0.2$. Case 2 with $a = 1$, $b = 3$, $\mu = 2$. Case 2 with $a = 1$, $b = 3$, $m = 2$. Case 4 with the empirical sample $\{1.0, 3.0, 4.0, 5.0, 5.5, 6.0, 6.5, 7.0, 7.5, 8.0\}$ and $\alpha = 0.95$.

Higher order moment information on X (e.g., correlation bounds) cannot be handled or processed yet. This is a current research field, cf., e.g., [48].

In Ferson et al. [49] we find an exhaustive description which construction techniques can be applied to construct a p-box, related to the type of available information. Moreover, it is illustrated how to construct p-boxes from different uncertainty models like DS structures (cf. Section 2.5) or Bayesian update estimates (cf. Section 2.4).

The studies on p-boxes have already lead to successful software implementations, cf. Berleant & Xie [12], Ferson [45].

To compute functions f of p-boxes, that means we have a p-box for $\varepsilon^1, \dots, \varepsilon^n$ and seek a p-box for $f = f(\varepsilon^1, \dots, \varepsilon^n)$, one first regards f consisting of elementary arithmetical operations and finds bounds for these expressions. To this end one discretizes the bounds for $\varepsilon^1, \dots, \varepsilon^n$ towards a discretization of the bounds for f, and then finds an expression for the bound of f in terms of the bounds for $\varepsilon^1, \dots, \varepsilon^n$. This can be done for all elementary arithmetic operations, without independence assumption for $\varepsilon^1, \dots, \varepsilon^n$, cf. Williamson & Downs [166], [167]. Thus the research on arithmetics for random variables actually builds the foundation of p-boxes. The dependency problem is not trivial, assume that one has independent random variables X, Y, Z, then the variables $S = X + Y$ and $T = Y \cdot Z$ are not independent in general.

One learns that the problem of rigorously quantifying probabilities given incomplete information – as done with probability arithmetic and p-boxes – is highly complex, even for simple problems, e.g., [90]. Due to their constructions the methods are rather restricted to lower dimensions and non-complex models f. Black box functions f cannot be handled as one requires knowledge about the involved arithmetic operations. All in all, they often appear not to be reasonably applicable in many real-life situations. On the other hand, as soon as we can apply methods like p-boxes to calculate with bounds on probability distributions, we are not restricted to the use of selecting less rigorous single distribution assumptions (e.g., maximum entropy) anymore.

Two more remarks about p-boxes: First, the definition of p-boxes can be generalized to higher dimensions based on the definition of higher dimensional CDFs, cf. Destercke et al. [32]. However, this has not lead to practical results yet. Second, probability arithmetic given only the information $X \in [a, b]$

can be regarded as a generalization of interval arithmetic. It is also related to the world of imprecise probabilities (cf. Section 2.9) via sets of measures. From a p-box $[F_l, F_u]$ for X one can infer bounds on the expectation for $f(X)$ by $\langle f \rangle_l = \inf_{F_l \leq F \leq F_u} \int_\Omega f dF$, $\langle f \rangle_u = \sup_{F_l \leq F \leq F_u} \int_\Omega f dF$, regarding $F_l \leq F \leq F_u$ as a set of measures, e.g., [160], [84]. The bounds can be computed numerically by discretization and formulation of a linear programming problem (LP), cf. Utkin & Destercke [157].

2.4 Bayesian inference

As soon as incomplete information is based on subjective knowledge and can be updated iteratively by additional information, one can consider using Bayesian inference to handle uncertainties.

Recall the definition of **conditional probability** given in Section 2.1.6: $\Pr(A \mid B) := \frac{\Pr(A \cap B)}{\Pr(B)}$. From this we get $\frac{\Pr(A|B)}{\Pr(B|A)} = \frac{\Pr(A \cap B) \cdot \Pr(A)}{\Pr(B) \cdot \Pr(B \cap A)}$. Since $\Pr(A \cap B)$ $= \Pr(B \cap A)$ we have

$$\Pr(A \mid B) = \frac{\Pr(B \mid A) \Pr(A)}{\Pr(B)}, \tag{2.44}$$

which is well-known as Bayes' rule [6]. Analogously to conditional probabilities we can define the **conditional density** of a random variable ε^1 conditional on ε^2 by $\rho(\varepsilon^1 \mid \varepsilon^2) = \frac{\rho(\varepsilon^1, \varepsilon^2)}{\rho(\varepsilon^2)}$, where $\rho(\varepsilon^1, \varepsilon^2)$ is the joint distribution of ε^1 and ε^2. Then the joint PDF of the random vector $\varepsilon = (\varepsilon^1, \varepsilon^2)$ can be represented by $\rho(\varepsilon) = \rho(\varepsilon^1 \mid \varepsilon^2)\rho(\varepsilon^2)$. Bayesian inference means reasoning on the basis of Bayes' rule which in some sense enables to invert causalities calculating with conditional probabilities.

For example, a statistician who is unfortunately a medical lay person wants to asses the probability that his child has a cold given that the child has sneezed 3 times in a row. All information he has is the probability of the event that the child sneezes 3 times in a row if it has a cold, so he wants to conclude from a symptom to make a diagnosis, but he just knows the probability of the symptoms given a diagnosis. Of course, this example is not representative as it lacks the complexity of real-life situations. The generalization for multiple possible diagnoses and symptoms is modeled by so-called Bayesian or belief networks (BNs). A BN is a directed acyclic graph (DAG) between states of a system and observables. A node $\mathcal{N}$ and its parent nodes in the DAG represent the input information of the network which consists of tables of conditional probabilities of $\mathcal{N}$ conditional on its parent nodes. The whole DAG represents the joint distribution of all involved variables, even in higher dimensional situations. Computations using BNs can be done efficiently on the DAG structure, assumed that all conditional probabilities are precisely known.

What if the conditional PDFs of the tables of conditional probabilities in BNs are unknown or not precisely known? This happens frequently in practice, in particular for variables conditional on multiple further variables. The Bayesian approach appears to become useless in this case. A generalized approach to BNs with imprecise probabilities can be studied on the basis of so-called **credal networks**, e.g., [25], [64]. A credal network is a set of BNs with the same DAG structure, but imprecise values in the conditional probability tables. The probabilities can be given as intervals, or more generally described.

The Bayesian approach accounts for uncertainties in statistical parameters like distribution parameters Θ, and criticizes the standard CDF concept to treat subjective expert opinions about Θ the same way as objective knowledge. However, it can be shown that the same mathematical mechanisms apply, so the involved calculations are mathematically equivalent: Classically, expectation of a function $f(X)$ is $\langle f\rangle_{\text{classic}} = \int f(x)\rho(x \mid \Theta)dx$, given Θ as a fixed distribution parameter $\rho(x \mid \Theta) = \rho_\Theta(x)$, cf. Section 2.1.1. Now consider Θ to be a random variable, so $\rho = \rho(x,\Theta)$ is the new joint PDF. Expectation now becomes $\langle f\rangle_{\text{Bayes}} = \int f(x)\rho(x,\Theta)dxd\Theta = \int f(x)\rho(x \mid \Theta)\rho(\Theta)dxd\Theta = \int \langle f\rangle_{\text{classic}}\rho(\Theta)d\Theta$.

In case of unknown joint distributions, conditional probabilities can be used to update subjective knowledge towards the actual joint CDF. Bayes' rule for densities reads

$$\rho(\varepsilon^1 \mid \varepsilon^2) = \frac{\rho(\varepsilon^2 \mid \varepsilon^1)}{\rho(\varepsilon^2)} \cdot \rho(\varepsilon^1). \tag{2.45}$$

Let $\rho(\varepsilon^1)$ be an initially assumed density. After obtaining additional information $\rho(\varepsilon^2 \mid \varepsilon^1)$, $\rho(\varepsilon^2)$ Bayes' rule serves to update $\rho(\varepsilon^1)$ to $\rho(\varepsilon^1 \mid \varepsilon^2)$ with update ratio $\frac{\rho(\varepsilon^2|\varepsilon^1)}{\rho(\varepsilon^2)}$. One calls $\rho(\varepsilon^1)$ the **prior** and $\rho(\varepsilon^1 \mid \varepsilon^2)$ the **posterior**.

Here we have the crucial problem, how to select the prior. For a reasonable choice real statistical data is needed in sufficient amount. Additionally, of course, incoming new observations are required for updating. Priors can be chosen, e.g., with the maximal entropy principle, cf. Section 2.1.3. In practice one often chooses a normal distribution to simplify calculations, or conjugate priors, i.e., a distribution where the posterior has a similar shape like the prior except from a change in some parameters. Actually, it is a well-known criticism that the choice of the prior often seems to be quite arbitrary and merely in the will of the statistician.

The Bayesian approach also applies in design optimization, cf. Zhou & Mourelatos [172]. Similar to RBDO (2.39) one minimizes a certain objective like design cost subject to probabilistic constraints involving the failure distribution. The associated joint distribution is estimated and updated from available data, starting with conjugate priors.

2.5 Dempster-Shafer theory

Dempster-Shafer theory enables to process incomplete uncertainty information allowing to compute bounds for failure probabilities and reliability.

We start with defining fuzzy measures, cf. Sugeno [153]. A **fuzzy measure** $\widetilde{\mu} : 2^\Omega \to [0,1]$, fulfills

$$\widetilde{\mu}(\emptyset) = 0, \widetilde{\mu}(\Omega) = 1, \tag{2.46}$$

$$A \subseteq B \Rightarrow \widetilde{\mu}(A) \leq \widetilde{\mu}(B). \tag{2.47}$$

The main difference to a probability measure is the absence of additivity. Instead, fuzzy measures only satisfy monotonicity (2.47). To find lower and upper bounds for an unknown probability measure given incomplete information one seeks two fuzzy measures **belief** Bel and **plausibility** Pl, where Bel is a fuzzy measure with $\text{Bel}(A \cup B) \geq \text{Bel}(A) + \text{Bel}(B) - \text{Bel}(A \cap B)$, and Pl is a fuzzy measure with $\text{Pl}(A \cup B) \leq \text{Pl}(A) + \text{Pl}(B) - \text{Pl}(A \cap B)$.

To construct the measures Bel and Pl from the given uncertainty information one formalizes the information as a so-called **basic probability assignment** $m : 2^\Omega \to [0,1]$ on a finite set $\mathcal{A} \subseteq 2^\Omega$ of non-empty subsets A of Ω, such that

$$m(A)\begin{cases} > 0 & \text{if } A \in \mathcal{A}, \\ = 0 & \text{otherwise,} \end{cases} \tag{2.48}$$

and the normalization condition $\sum_{A \in \mathcal{A}} m(A) = 1$. Sometimes m is also called **basic belief assignment**.

The basic probability assignment m is interpreted as the exact belief focussed on A, and not in any strict subset of A. The sets $A \in \mathcal{A}$ are called **focal sets**. The structure $(m, \mathcal{A})$, i.e., a basic probability assignment together with the related set of focal sets, is called a **Dempster-Shafer structure** (DS structure).

Given a DS structure $(m, \mathcal{A})$ we can construct Bel and Pl by

$$\mathrm{Bel}(B) = \sum_{\{A \in \mathcal{A} \mid A \subseteq B\}} m(A), \tag{2.49}$$

$$\mathrm{Pl}(B) = \sum_{\{A \in \mathcal{A} \mid A \cap B \neq \emptyset\}} m(A) \tag{2.50}$$

for $B \in 2^{\Omega}$.

Thus Bel and Pl have the sought property Bel $\leq$ Pr $\leq$ Pl by construction and, moreover, satisfy $\mathrm{Bel}(B) = 1 - \mathrm{Pl}(B^c)$. The information contained in the two measures Bel and Pl induced by the DS structure is often called a **random set**.

In the classical case the additivity of non-fuzzy measures would yield $\mathrm{Pl}(B) = 1 - \mathrm{Pl}(B^c) = \mathrm{Bel}$. Thus Bel = Pr = Pl and classical probability theory becomes a special case of DS theory. Also note that if we have a DS structure on the singletons of a finite Ω, then we have full stochastic knowledge equivalent to a CDF.

DS structures can be obtained from expert knowledge or in lower dimensions from histograms, or from the Chebyshev inequality $\Pr(|X - \mu| \leq r) > 1 - \frac{\sigma^2}{r^2}$ given expectation value μ and variance σ^2 of a random variable X, cf. [123], [124], [125]: Let $r = \frac{\sigma}{\sqrt{1-\alpha}}$ for a fixed confidence level α, then $\Pr(\{|X - \mu| \leq \frac{\sigma}{\sqrt{1-\alpha}}\}) > \alpha$. The sets $C_\alpha := \{\omega \in \Omega \mid |X(\omega) - \mu| \leq \frac{\sigma}{\sqrt{1-\alpha}}\}$

for different values of α define focal sets, and we get Belief and Plausibility measures by $\text{Bel}(C_\alpha) = \alpha$ and $\text{Pl}(C_\alpha^c) = 1 - \alpha$, respectively.

To extend one-dimensional focal sets to the multi-dimensional case one can generate joint DS structures from the Cartesian product of marginal basic probability assignments assuming random set independence, cf. Couso et al. [24], or from weighting the 1-dimensional marginal focal sets, cf. Fetz [51]. In [172] we find the suggestion to employ Bayesian techniques to estimate and update DS structures from little amount of information.

To combine different, or even conflicting DS structures $(m_1, \mathcal{A}_1)$, $(m_2, \mathcal{A}_2)$ (in case of multiple bodies of evidence, e.g., several different expert opinions) to a new basic probability assignment m_{new} one uses Dempster's rule of combination [28], forming the basis of **Dempster-Shafer theory** or **evidence theory** [144],

$$m_{\text{new}}(B) = \sum_{\{A_1 \in \mathcal{A}_1, A_2 \in \mathcal{A}_2 | A_1 \cap A_2 = B\}} \frac{m_1(A_1) m_2(A_2)}{K} \tag{2.51}$$

with the normalization constant $K = 1 - \sum_{\{A_1 \in \mathcal{A}_1, A_2 \in \mathcal{A}_2 | A_1 \cap A_2 = \emptyset\}} m_1(A_1) m_2(A_2)$ which is interpreted as the **conflict**.

The combination rule enables to compute a joint DS structure. Also note that the combination rule is a generalization of Bayes' rule, motivated by the criticism that a single probability assignment cannot model the amount of evidence one has.

The complexity of the rule is strongly increasing in higher dimensions, and in many cases requires independence assumptions for simplicity reasons avoiding problems with interacting variables. It is not yet understood how the dimensionality issue can be solved. Working towards more efficient computational implementations of evidence theory it can be attempted to decompose the high-dimensional case in lower dimensional components which leads to so-called compositional models, cf. Jirousek et al. [75].

The **extension** of a function f is based on the joint DS structure $(m, \mathcal{A})$. The new focal sets of the extension are $B_i = f(A_i)$, $A_i \in \mathcal{A}$, the new basic probability assignment is $m_{\text{new}}(B_i) = \sum_{\{A_i \in \mathcal{A} | f(A_i) = B_i\}} m(A_i)$.

To embed DS theory in design optimization one formulates a constraint on the upper bound of the failure probability p_f which should be smaller than an admissible failure probability p_a, i.e., $\mathrm{Pl}(\mathbb{F}) \leq p_a$, for the failure set $\mathbb{F}$. This can be studied in Mourelatos & Zhou [110] as evidence based design optimization (EBDO). One can also find further direct applications in engineering computing, e.g., in [53], [123].

DS structures enable to construct p-boxes [13], [32], [157], i.e., to determine lower bounds F_l and upper bounds F_u of the CDF of a random variable X,

$$\begin{aligned} F_l(t) &= \mathrm{Bel}(\{\omega \in \Omega \mid X(\omega) \leq t\}), \\ F_u(t) &= \mathrm{Pl}(\{\omega \in \Omega \mid X(\omega) \leq t\}). \end{aligned}$$

Conversely it is possible to generate a DS structure that approximates a given p-box discretely, cf. [3], [30], [49]. Fix some levels $\alpha_1 \leq \alpha_2 \leq \cdots \leq \alpha_N = 1$ of the p-box, then generate focal sets by

$$\begin{aligned} A_i &:= [\inf\{x \mid F_u(x) = \alpha_i\}, \inf\{x \mid F_l(x) = \alpha_i\}], \\ m(A_1) &= \alpha_1, m(A_i) = \alpha_i - \alpha_{i-1}, i = 2, \dots, N. \end{aligned} \tag{2.52}$$

Another relation to a different uncertainty representation concerns nested focal sets, i.e., $\mathcal{A} = \{A_1, A_2, \dots, A_m\}$, $A_1 \subseteq A_2 \subseteq \cdots \subseteq A_m$. In this case

$$\mathrm{Bel}(A \cap B) = \min(\mathrm{Bel}(A), \mathrm{Bel}(B)), \tag{2.53}$$

$$\mathrm{Pl}(A \cup B) = \max(\mathrm{Pl}(A), \mathrm{Pl}(B)). \tag{2.54}$$

For nested focal sets the fuzzy measures Bel and Pl directly correspond to possibility and necessity measures, respectively, which appear in fuzzy set theory, cf. Dubois & Prade [34], as we will see in the next section.

We have learned that DS structures can unify several different uncertainty models, see, e.g., [86], but cannot overcome the curse of dimensionality being prohibitively expensive in higher dimensions.

2.6 Fuzzy sets

The development of fuzzy sets has started roughly in parallel to the development of DS theory with the goal to model vague verbal descriptions in absence of any statistical data. It is a generalization of conventional set theory redefining the characteristic function of a set A by a so-called **membership function** μ_A. The value $\mu_A(x)$ indicates the membership value of an uncertain variable x with respect to A. The value can be any real number between 0 and 1 as opposed to the characteristic function $1_A(x)$ which only takes binary values. A **fuzzy set** is a set A together with its related membership function μ_A.

This section will give a short overview on fuzzy sets, focussing on their application for uncertainty handling. The following terms play an important role in the theory of fuzzy sets. The **height** h of a fuzzy set is defined by $h := \max_x \mu_A(x)$. The **support** of a fuzzy set is the set $\{x \mid \mu_A(x) \neq 0\}$. The **core** or **modal values** of a fuzzy set is the set $\{x \mid \mu_A(x) = 1\}$. The α**-cut** C_α of a fuzzy set for a fixed value $\alpha \in [0, 1]$ is the set

$$C_\alpha := \{x \mid \mu_A(x) \geq \alpha\}. \tag{2.55}$$

The α-cut is determined by the values of the membership function. Conversely one can construct μ_A from the knowledge of the α-cuts, cf. Zadeh [170], to achieve an α-cut based representation of a fuzzy set:

$$\mu_A(x) = \sup_\alpha \min(\alpha, 1_{C_\alpha}(x)). \tag{2.56}$$

Note the relationship between BPA-structures on nested focal sets, cf. Section 2.5, and α-cuts of a fuzzy set with non-empty core, which are nested by definition, i.e., $C_\alpha \subseteq C_\beta$ for $\alpha \geq \beta$. Let $1 = \alpha_1 \geq \alpha_2 \geq \cdots \geq \alpha_N = 0$ be α-levels of a fuzzy set, then we can construct a BPA m on the α-cuts C_{α_i} by $m(C_{\alpha_i}) = \alpha_i - \alpha_{i+1}$, $i < N$, $m(C_{\alpha_N}) = \alpha_N$. Conversely a BPA-structure on nested focal sets $A_1 \subseteq A_2 \subseteq \cdots \subseteq A_N$ allows to construct a fuzzy set by $\alpha_N = m(A_N)$, $C_{\alpha_N} = A_N$, $\alpha_{N-1} = m(A_N) + m(A_{N-1})$, $C_{\alpha_{N-1}} = A_{N-1}$, $\ldots$, $\alpha_1 = \sum_{i=1}^N m(A_i) = 1$, $C_{\alpha_1} = A_1$, and then applying (2.56). Thus it is possible to convert expert knowledge modeled by a fuzzy set into a DS structure. Using the Dempster's rule, however, to combine different bodies of evidence in general leads to non-nested focal sets, hence a conversion back to the fuzzy set formalism is not possible after applying a combination rule.

Some special cases of fuzzy sets motivated the notation of fuzzy intervals and fuzzy numbers, cf. Zadeh [169]. A **fuzzy interval** or **convex fuzzy set** is a fuzzy set with $\mu_A(x) \geq \min(\mu_A(a), \mu_A(b))$ for all $a, b,\ x \in [a, b]$. A **fuzzy number** is a fuzzy interval with closed α-cuts, compact support, and a unique modal value.

Let μ_A, μ_B be the membership functions of the fuzzy sets A and B, respectively. Then we get the following properties, arising from fuzzy logic,

$$\begin{aligned} \mu_{A\cap B}(x) &= \min(\mu_A(x), \mu_B(x)), \\ \mu_{A\cup B}(x) &= \max(\mu_A(x), \mu_B(x)), \\ \mu_{A^c}(x) &= 1 - \mu_A(x). \end{aligned} \tag{2.57}$$

Recall the notation of a fuzzy measure, cf. Section 2.5. For all fuzzy measures μ obviously holds $\mu(A\cap B) \leq \min(\mu(A), \mu(B))$ and $\mu(A\cup B) \geq \max(\mu(A), \mu(B))$. From these properties we define 2 special, limiting cases of fuzzy measures, cf. [34], [171], the simplicity of which makes them computationally attractive. A **possibility measure** is a fuzzy measure Pos with

$$\text{Pos}(A \cap B) = \min(\text{Pos}(A), \text{Pos}(B)). \tag{2.58}$$

A **necessity measure** is a fuzzy measure Nec with

$$\text{Nec}(A \cup B) = \max(\text{Nec}(A), \text{Nec}(B)). \tag{2.59}$$

Note that possibility and necessity measures directly correspond to plausibility and belief measures of DS structures in case of nested focal sets, cf. Section 2.5, and can thus be regarded as lower and upper probability measure bounds.

A **possibility distribution** is a possibility measure defined on the singletons of a finite Ω. By the equations (2.57) we see that a membership function of a fuzzy set with non-empty core can be interpreted as a possibility distribution. A possibility distribution Π imposes an associated possibility measure by $\text{Pos}(A) = \max_{x\in A} \Pi(x)$.

The definition of a fuzzy set and its membership function in higher dimensions is a straightforward generalization of the one-dimensional case. The extension of a function $f(x) = z$, $f : \mathbb{R}^n \to \mathbb{R}$, for a fuzzy set with membership function μ is constructed by the **extension principle** for a new membership function

$$\mu_{\text{new}}(z) = \sup_{x \in f^{-1}(z)} \mu(x), \tag{2.60}$$

cf. Zadeh [169]. The construction involves an optimization problem with rapidly increasing complexity in higher dimensions.

It can be attempted to solve this problem by reduction of the problem to the α-cuts of the fuzzy set, cf. Section 2.6.1, or by sensitivity analysis, cf. Section 2.1.7.

Except from the dimensionality issue another criticism of fuzzy sets is the fact that the assignment of membership functions appears to be quite arbitrary, often defined by a single expert opinion. In lower dimensions membership functions can be estimated, e.g., from histograms, but there is no general, statistically well-grounded basis for the assignment of membership functions. Of course, if only vague verbal descriptions, i.e., highly informal uncertainty information, is available statistical properties are entirely absent. In this case, which represents the classical motivation of fuzzy sets, it can be argued that it is impossible to formulate a general recipe for processing the information. However, usually the information consists of a mixture of statistical and fuzzy descriptions, and conventional fuzzy methods cannot combine both. The concept of fuzzy randomness, cf. [92], [102], [106], [133], is one attempt of a combination.

The applications of fuzzy methods in engineering computing are vast. A famous application of fuzzy methods is fuzzy control, cf. Sugeno [154]. Moreover, most design analyzing methods have their counterparts in the context of fuzzy sets, for instance, fuzzy reliability methods (e.g., [21], [106]), fuzzy differential equations (e.g., [52]), fuzzy finite element methods (e.g., [50], [103], [111]), fuzzy ARMA and other stochastic processes (e.g., [104]).

In fuzzy statistics, i.e., with sample points that are modeled as fuzzy numbers, one can apply statistical methods on non-precise data, cf. Viertl [158].

In design optimization fuzzy methods can be used to find clusters of permissible designs with fuzzy clustering methods, e.g., [70], [77]. Seeking the optimal design one can use fuzzy methods to compare different design points of different clusters with respect to some criterion, e.g., weighted distances from design constraints [7], [20], [73].

Also, similar to RBDO and EBDO, the concept of possibility measures can be used to formulate safety constraints of a design optimization problem (possibility theory based design optimization, PBDO), cf. Mourelatos & Zhou [109].

The following subsection presents a special fuzzy set based method which is highlighted because of its relationship to our approach based on clouds, cf. Section 2.8.

2.6.1 α-level optimization

The α-level optimization approach Möller et al. [105] is the most relevant fuzzy set based method for our purposes as it applies also in higher dimensional real-life situations and uses similar techniques as we will use employing the clouds formalism.

The α-level optimization method combines the α-cut representation (2.56) and the extension principle to determine the membership function μ_f of a function $f(\varepsilon)$, $f : \mathbb{R}^n \to \mathbb{R}$, given the membership function μ of the variable ε. This is achieved by constructing the α-cuts $C_{f_{\alpha_i}}$ belonging to μ_f from the α-cuts C_{α_i} belonging to μ. To this end one solves the optimization problems

$$\min_{\varepsilon \in C_{\alpha_i}} f(\varepsilon), \tag{2.61}$$

$$\max_{\varepsilon \in C_{\alpha_i}} f(\varepsilon) \tag{2.62}$$

for different discrete values α_i. Finally from the solution f_{i_*} of (2.61) and f_i^* of (2.62) one constructs the α-cuts belonging to $f(\varepsilon)$ by $C_{f_{\alpha_i}} = [f_{i_*}, f_i^*]$.

To simplify the optimization step one assumes sufficiently nice behaving functions f and computationally nice fuzzy sets, i.e., convex fuzzy sets, typically triangular shaped fuzzy numbers.

In n dimensions one optimizes over a hypercuboid, obtained by the Cartesian product of the α-cuts $C_{\alpha_i} = C_{\alpha_i}^1 \times C_{\alpha_i}^2 \times \cdots \times C_{\alpha_i}^n$, where $C_{\alpha_i}^j := \{\varepsilon^j \mid \mu^j(\varepsilon^j) \geq \alpha_i\}$, $\mu^j(\varepsilon^j) := \sup_{\varepsilon^k, k \neq j} \mu(\varepsilon)$, $\varepsilon = (\varepsilon^1, \varepsilon^2, \dots, \varepsilon^n)$. Here one has to assume non-interactivity of the uncertain variables $\varepsilon^1, \dots, \varepsilon^n$.

Using a discretization of the α-levels by a finite choice of α_i the computational effort for this methods becomes tractable. From (2.56) one gets a step function for μ_f which is usually linearly approximated through the points f_{i*} and f_i^* to generate a triangular fuzzy number.

2.7 Convex methods

Convex methods model uncertainty by so-called **anti-optimization** over convex sets, cf. Ben-Haim & Elishakoff [8], [38]. Assume that we wish to find the design point $\theta = (\theta^1, \theta^2, \dots, \theta^{n_o})$ with the minimal design objective function value $g(\theta, \varepsilon)$, $g : \mathbb{R}^{n_o} \times \mathbb{R}^n \to \mathbb{R}$ under uncertainty of the vector of input variables ε. Also assume that the uncertainty of ε is described by a convex set $\mathcal{C}$. Anti-optimization means finding the worst-case scenario for a fixed design point θ by the solution of an optimization problem of the type

$$\begin{aligned} \max_{\varepsilon} \quad & g(\theta, \varepsilon) \\ \text{s.t.} \quad & \varepsilon \in \mathcal{C} \end{aligned} \tag{2.63}$$

The corresponding design optimization problem would be

$$\begin{aligned} \min_{\theta} \quad \max_{\varepsilon} \quad & g(\theta, \varepsilon) \\ \text{s.t.} \qquad\qquad & \varepsilon \in \mathcal{C} \\ & \theta \in T \end{aligned} \tag{2.64}$$

where T is the set of possible selections for the design θ. As the inner level of problem of (2.64), i.e., equation (2.63), maximizes the objective which is sought to be minimized for the design optimization in the outer level (i.e., one seeks the design with minimal worst-case), the term *anti-optimization* has been proposed for this approach. We will revisit these types of optimization problems in more detail in Chapter 4 and Chapter 5.

Investigating convex regions for the worst-case search is motivated by the fact that in many cases the level sets of probability densities are convex sets, e.g., ellipsoids for normal distributions. In this respect the term convex uncertainty for a random vector $\varepsilon \in \mathbb{R}^n$ is characterized by a convex set $\mathcal{C} = \{\varepsilon \mid Q(\varepsilon) \leq c\}$, where Q is a quadratic form and ε is known to belong to $\mathcal{C}$ with some confidence. The quadratic form could be, e.g., $Q(\varepsilon) = (\varepsilon - m)^T C^{-1}(\varepsilon - m)$ with a vector of nominal values m and an estimated covariance matrix C.

Once one has a description by convex uncertainty one can apply optimization methods which can make convex methods applicable even in higher dimensions.

It should be remarked that this particular idea is one of the inspirations for the potential clouds concept, see the next section and Section 3.2, where the potential function will be constructed to have convex level sets.

2.8 Potential clouds

This section will give an overview on uncertainty representation by means of clouds, introduced in Neumaier [116], inspired by and combining ideas from p-boxes, random and fuzzy sets, convex methods, interval and optimization methods. Before we go into the details in Chapter 3 we will learn in this section about the intuitive approach that has lead to use clouds for uncertainty modeling, handling incomplete information in higher dimensions, and to weave the methodology into an optimization problem formulation.

The goal is to construct confidence regions in which we should be able to search for worst-case scenarios via optimization techniques. The construction should be possible on the basis of scarce, high-dimensional data, incomplete information, unformalized knowledge and information updates. As mentioned in previous sections, in lower dimensions and provided real empirical data one has powerful tools, like KS, e.g., to bound the CDF of a random variable X. What could one do to tackle the same problems for higher dimensional random vectors $\varepsilon \in \mathbb{R}^n$ with little or no information available? To generate data we will first simulate a data set and modify it with respect to the available uncertainty information. To reduce the dimensionality of the problem we will use a potential function $V : \mathbb{R}^n \to \mathbb{R}$. We will bound the CDF of $V(\varepsilon)$ using KS as in the one-dimensional case (like a p-box on $V(\varepsilon)$, cf. Section 2.3). From the bounds on the CDF of $V(\varepsilon)$ we get lower and upper confidence regions for

$V(\varepsilon)$, and finally lower and upper confidence regions for ε as level sets of V (similar to α-cuts of an interval valued fuzzy set, i.e., a fuzzy set with interval valued membership function – however, put in a probabilistic context).

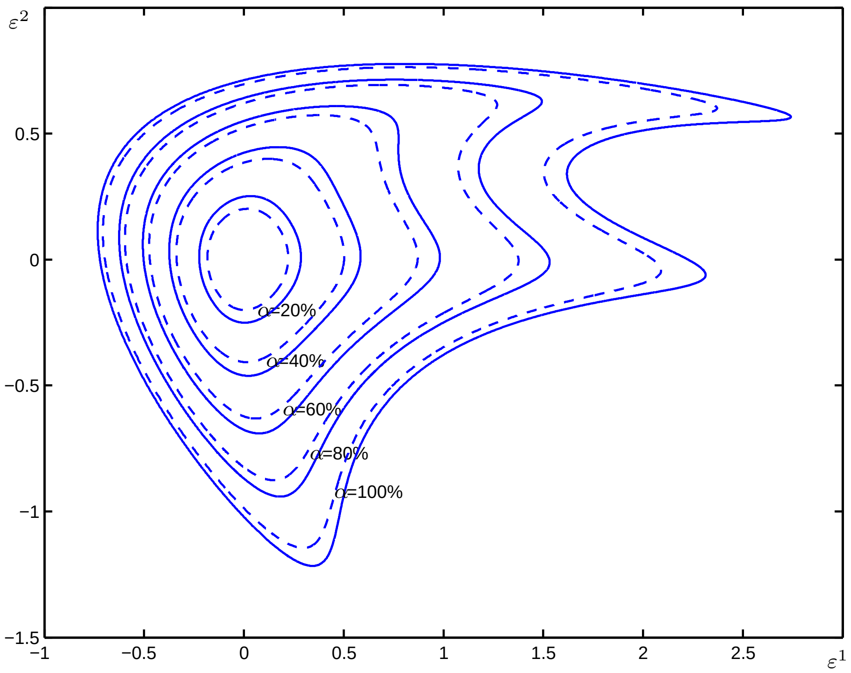


Figure 2.5: Nested confidence regions for the example of a 2-dimensional potential cloud, $\alpha = 0.2, 0.4, 0.6, 0.8, 1$. The lower confidence regions $\underline{C}_\alpha$ plotted with dashed lines, the upper confidence regions $\overline{C}_\alpha$ with solid lines.

We denote the lower and upper confidence regions by $\underline{C}_\alpha$ and $\overline{C}_\alpha$, respectively, with the confidence level α. This nested collection of confidence regions is a **potential cloud**, cf. Figure 2.5.

Note that potential clouds extend the p-box concept to the multivariate case without the exponential growth of work in the conventional p-box approach. From the fact that we construct a p-box on $V(\varepsilon)$ one can also see the relation to DS structures generated from p-boxes as in (2.52), with $A_i = \overline{C}_{\alpha_i} \backslash \underline{C}_{\alpha_i}$. Thus the focal sets are determined by the level sets of V.

The potential clouds approach not only helps us to overcome the curse of dimensionality in real-life applications, but also it turns out to enable a flexible uncertainty representation. It can process incomplete knowledge of different kinds and allows for an adaptive interaction between the uncertainty elicitation and the optimization phase, reducing the incompleteness of epistemic information via information updating. This is a unique feature in higher dimensions.

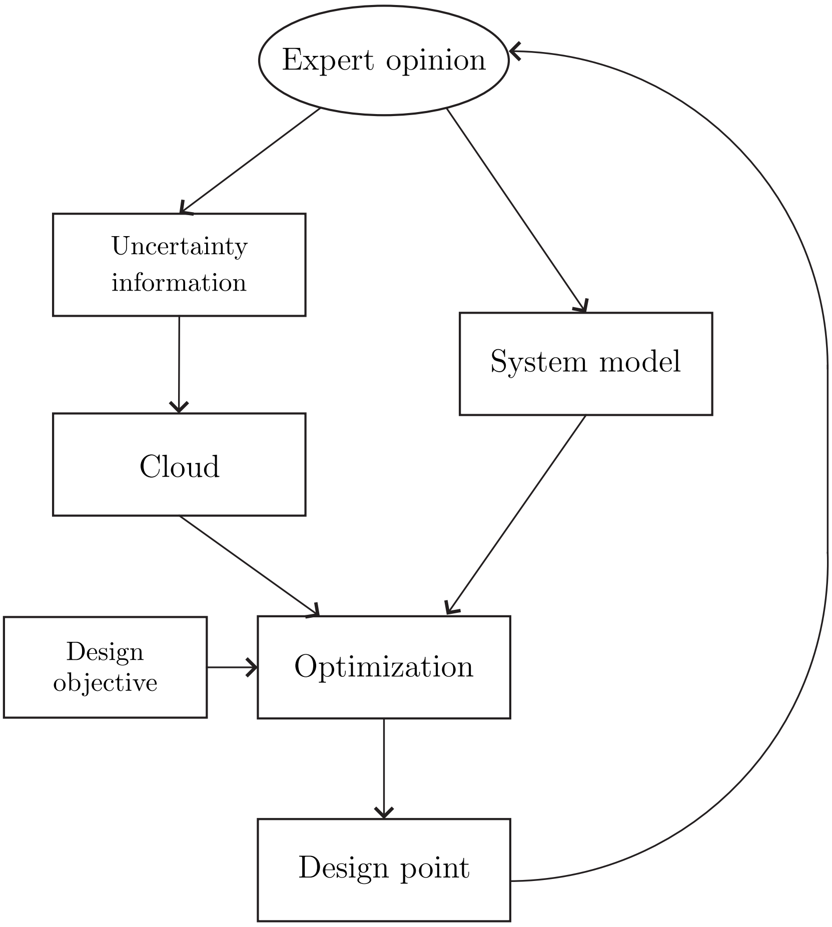


Figure 2.6: Basic concept.

Figure 2.6 illustrates the basic concept of the approach. The designing expert provides an underlying system model – e.g., given as a black box model – and all currently available uncertainty information on the input variables of the model. The information is processed to generate a cloud that provides a nested collection of regions of relevant scenarios parameterized by a confidence level α. Thus we produce safety constraints for the optimization. The optimization minimizes a certain objective function (e.g., cost, mass) subject to the safety constraints to account for the robustness of the design, and subject to the functional constraints which are represented by the system model. The results of the optimization, i.e., the automatically found optimal design point and the worst-case analysis, are returned to the expert, who is given an interactive possibility to provide additional uncertainty information afterwards and rerun the procedure.

2.9 Imprecise probabilities

Last, but not least, we mention the family of concepts from imprecise probabilities, cf. Walley [159]. Imprecise probabilities can be regarded as the most general point of view describing scarce, vague, incomplete, or conflicting uncertainty information. This approach alludes to existing uncertainty models being not sufficiently general to handle all kinds of uncertainty, and it encourages to develop a **unified formulation** [161], opposed to the toolbox philosophy mentioned in Chapter 1.

In a nutshell, imprecise probabilities should not be a new tool, or replace existing tools, but rather complement classical concepts to a complete framework to handle more general situations. The family of imprecise probabilities includes upper and lower probabilities, cf. Smith [151], sets of probability measures, cf. Good [61], and many concepts introduced in the last sections appear as special cases, e.g., evidence theory, possibility measures.

2.10 Other approaches

The summary of uncertainty methods in the last sections represents a collection of most of the popular methods applied in real-life uncertainty modeling. Of course, it can be extended to a vast family of related approaches. In this section we briefly mention a few of them.

Chaotic probability models try to build an alternative framework to the standard probabilistic methods. They attempt to involve both subjective and objective uncertainties. The present stage of this theory still has a rather conceptual than applied nature. It can be studied, e.g., in [54], [76], [135]. In the development of a logic programming language with semantics to represent qualitative and quantitative knowledge about uncertainty one can study probabilistic logic programming, e.g., [140].

Now we turn our focus back to the uncertainty model of our choice which leads us to the next chapter: *clouds*.

Chapter 3

Uncertainty modeling with clouds

This chapter is dedicated to the study of **clouds**. After the short intuitive introduction in Section 2.8, we will now go into the details of this concept towards handling higher dimensional, incomplete uncertainty information.

At first we follow the clouds formalism from NEUMAIER [116] in Section 3.1, picking those ideas that are most attractive for the handling of high-dimensional uncertainties and elaborating them. Afterwards we investigate the special case of potential clouds in Section 3.2. We will see how they help to cope with dimensionality issues. In Section 3.3 we will learn how to interpret approximations and bounds on cumulative distribution functions in terms of clouds. Some considerations about suitable potential functions can be found in Section 3.4. A survey on the algorithms that realize the described uncertainty handling methods is given in Section 3.5.

3.1 The clouds formalism

We start with the formal definition of clouds and introduce the notation. Afterwards we restate the central results from [116], Sections 3 and 4, that will be relevant for our studies later on.

Let ε be an n-dimensional random vector, $\mathbb{M} := \varepsilon(\Omega)$. In the context of worst-case analysis we will sometimes call a particular value for the realization of ε an uncertain **scenario**.

A **cloud** is a mapping $\chi : \mathbb{M} \to \mathbb{I}[0,1]$, $\chi(x) = [\underline{\chi}(x), \overline{\chi}(x)]$, where $\mathbb{I}[0,1]$ denotes the set of closed and bounded intervals in $[0,1]$, $\chi(x)$ nonempty for all $x \in \mathbb{M}$, and

$$]0,1[\subseteq \bigcup_{x\in\mathbb{M}} \chi(x) \subseteq [0,1]. \tag{3.1}$$

We call $\underline{\chi}$ the **lower level** and $\overline{\chi}$ the **upper level** of the cloud, and denote the **width** of the cloud χ at x by $\overline{\chi}(x) - \underline{\chi}(x)$. A cloud is called **thin** if it has width 0 for all x.

We say that a random vector ε **belongs to a cloud** χ over $\mathbb{M}$, if

$$\Pr(\overline{\chi}(\varepsilon) \leq y) \leq y \leq \Pr(\underline{\chi}(\varepsilon) < y) \tag{3.2}$$

for all $y \in [0,1]$.

3.1.1 Remark. Assume that the 1-dimensional random variable X has the given CDF F, and F, $\underline{\chi}$ and $\overline{\chi}$ are continuous, invertible functions. Then equation (3.2) is equivalent to $F(\overline{\chi}^{-1}(y)) \leq y \leq F(\underline{\chi}^{-1}(y))$ which directly translates into the intuitive interpretation of lower and upper confidence regions, cf. Figure 2.5.

A cloud is called **discrete** if $\underline{\chi}$ and $\overline{\chi}$ only take finitely many different values, i.e., $\chi(x)$, $x \in \mathbb{M}$, only takes finitely many different values in $\mathbb{I}[0,1]$. It can be shown that discrete clouds can be constructed from samples of discrete or discretized continuous probability distributions in low dimensions via histograms, cf. [116, Theorem 3.1], and in principle can approximate arbitrary distributions arbitrarily well. As histograms and the related discrete clouds are having computational problems in higher dimensions, we define continuous clouds that will be much more important for our purposes: A cloud is called **continuous** if the lower level $\underline{\chi}$ and the upper level $\overline{\chi}$ are continuous functions.

There exists a close relationship between thin continuous 1-dimensional clouds and CDFs of real univariate random variables, cf. [116, Proposition 4.1].

3.1.2 Proposition. *Let $F(t) = \Pr(X \leq t)$ be the CDF of the random variable X, then $\chi(t) := F(t)$ defines a thin cloud and X belongs to χ, i.e., for the thin case* $\Pr(\chi(t) \leq y) = y, y \in [0,1]$, *if t has the same distribution as X.*

Proof. From the definition of a CDF it follows that $\chi := F$ satisfies the conditions that define a cloud. Due to the fact that $F(t)$ is uniformly distributed if t has the same CDF as X we have $\Pr(\chi(t) \leq y) = \Pr(F(t) \leq y) = y$. □

3.1.3 Remark. By Proposition 3.1.2 one sees that the information contained in a thin cloud represents full stochastic information. A cloud with $\underline{\chi} \equiv 0$ is called **fuzzy**. The information contained in a fuzzy cloud is equivalent to that of a fuzzy set with membership function $\overline{\chi}(x)$.

CDFs are well known from probability theory. In particular the univariate case is very handy, computationally appealing and intuitively understandable. However, we want to deal with significantly higher dimensions than 1. This leads to the idea to construct continuous clouds from user-defined potential functions $V : \mathbb{M} \to \mathbb{R}$.

3.1.4 Remark. Comparing the notation we introduced in this section with [116] it should be remarked that ours corresponds to the so-called **mirror cloud** $\chi' = 1-\chi$. This is unproblematic in view of [116] Corollary 2.2. Observe that the same set of random variables can be characterized by different clouds.

3.2 Potential clouds

As we learned in the last section potential function based clouds, in short **potential clouds**, are a special class of continuous clouds supposed to help to cope with high-dimensional uncertainties. Let ε be an n-dimensional random vector. Though the probability density of ε usually cannot be estimated for high n due to a lack of available data, the random variable $V := V(\varepsilon)$ is univariate and its CDF can be easily approximated by an empirical CDF. The idea is to construct a cloud from an interval-valued function χ of a user-defined potential function V, i.e., $\chi \circ V : \mathbb{M} \to \mathbb{I}[0,1]$. How this potential can be constructed in practice will be discussed in Section 3.4.

3.2.1 Theorem. *Suppose that $x \in \mathbb{M}$, $\underline{\alpha}$ continuous from the left and monotone, $\overline{\alpha}$ continuous from the right and monotone, $a \in \mathbb{R}$, V bounded below, and*

$$\underline{\alpha}(a) \leq \Pr(V(\varepsilon) < a), \tag{3.3}$$

$$\overline{\alpha}(a) \geq \Pr(V(\varepsilon) \leq a). \tag{3.4}$$

Then the mapping χ defined by

$$\chi(x) := [\underline{\alpha}(V(x)), \overline{\alpha}(V(x))], \tag{3.5}$$

defines a cloud and ε belongs to χ.

Proof. Obviously (3.1) holds for χ. One has to show (3.2). Let $y \in [0, 1]$. Assume that $\exists x \in \mathbb{R}$ with $\overline{\alpha}(x) = y$.

$$\begin{aligned} \Pr(\overline{\chi}(\varepsilon) \leq y) &= \Pr(\overline{\alpha}(V(\varepsilon)) \leq y) && \text{see (3.5)} \\ &= \Pr(\overline{\alpha}(V(\varepsilon)) \leq \overline{\alpha}(x)) && \text{definition of } x \\ &= \Pr(V(\varepsilon) \leq x) && \text{as } \overline{\alpha} \text{ is monotone} \\ &\leq \overline{\alpha}(x) \\ &= y \end{aligned}$$

If $\nexists x \in \mathbb{R}$ with $\overline{\alpha}(x) = y$ and $\overline{\alpha}(x) > y$ for all x, then

$$\begin{aligned} \Pr(\overline{\chi}(\varepsilon) \leq y) &= \Pr(\overline{\alpha}(V(\varepsilon)) \leq y) \\ &= 0 \leq y \end{aligned}$$

Otherwise if $\nexists x \in \mathbb{R}$ with $\overline{\alpha}(x) = y$, then $\exists y' \in [0, 1], x, h \in \mathbb{R}$ with $\overline{\alpha}(x + h) = y' > y$, and $\overline{\alpha}(x) \leq y$ as $\overline{\alpha}$ is continuous from the right. This yields

$$\begin{aligned} \Pr(\overline{\chi}(\varepsilon) \leq y) &= \Pr(\overline{\alpha}(V(\varepsilon)) \leq y) \\ &\leq \Pr(\overline{\alpha}(V(\varepsilon)) \leq \overline{\alpha}(x + h)) \\ &= \Pr(V(\varepsilon) \leq x + h) \\ &\leq \overline{\alpha}(x + h), \end{aligned}$$

thus in the limit $h \to 0$, $\Pr(\overline{\chi}(\varepsilon) \leq y) \leq y$.

Hence $\Pr(\overline{\chi}(\varepsilon) \leq y) \leq y$. We get $\Pr(\underline{\chi}(\varepsilon) < y) \geq y$ analogously, so χ fulfills (3.2). □

Usually the CDF of $V(\varepsilon)$ is unknown. However, Theorem 3.2.1 tells us we only need to find a lower bound $\underline{\alpha}(y) \leq \Pr(V(\varepsilon) < y)$ and upper bound $\overline{\alpha}(y) \geq \Pr(V(\varepsilon) \leq y)$, $\underline{\alpha}$, $\overline{\alpha}$ continuous from the right and monotone, to construct a potential cloud.

3.2.2 Remark. This implies the relation between potential and p-boxes mentioned in Section 2.8. A potential cloud can be constructed from a p-box on $V(\varepsilon)$ regarding it as a random variable.

3.2.3 Remark. The constructed cloud χ is thin if V has a known continuous CDF: Let F be the CDF of $V(\varepsilon)$, then

$$\underline{\alpha}(y) = \Pr(V(\varepsilon) \leq y) = \Pr(V(\varepsilon) < y) = \overline{\alpha}(y) = F(y)$$

and $\chi(x) = [F(V(x)), F(V(x))]$.

The last remark leads us to an important interpretation in terms of confidence regions for ε. Let $\alpha \in [0, 1]$ be a given confidence level, let F be the CDF of $V(\varepsilon)$ and precisely known. Then $C_\alpha := \{x \in \mathbb{M} \mid F(V(x)) \leq \alpha\}$ is an α-confidence region for ε, as $\Pr(\varepsilon \in C_\alpha) = \Pr(F(V(\varepsilon)) \leq \alpha) = \alpha$.

More importantly, Theorem 3.2.1 tells us that it is sufficient to find an appropriate bounding $\underline{\alpha}$, $\overline{\alpha}$ on F to construct a potential cloud! As shown in Section 2.1.3, this can be achieved approximately (at a given confidence level) by KS statistics. Similarly to C_α we define the **lower α-cut**

$$\underline{C}_\alpha := \begin{cases} \{x \in \mathbb{M} \mid V(x) \leq \underline{V}_\alpha\} & \text{if } \underline{V}_\alpha := \min\{V_\alpha \in V(\mathbb{M}) \mid \overline{\alpha}(V_\alpha) = \alpha\} \text{ exists,} \\ \emptyset & \text{otherwise.} \end{cases} \tag{3.6}$$

Analogously the **upper α-cut**

$$\overline{C}_\alpha := \begin{cases} \{x \in \mathbb{M} \mid V(x) \leq \overline{V}_\alpha\} & \text{if } \overline{V}_\alpha := \max\{V_\alpha \in V(\mathbb{M}) \mid \underline{\alpha}(V_\alpha) = \alpha\} \text{ exists,} \\ \mathbb{M} & \text{otherwise.} \end{cases} \tag{3.7}$$

The sets $\underline{C}_\alpha$, $\overline{C}_\alpha$ are the protagonists in the following important result for potential clouds.

3.2.4 Theorem. *The regions $\underline{C}_\alpha$ and $\overline{C}_\alpha$ can be interpreted as a nested collection of lower and upper confidence regions, i.e.,*

1. *The region $\underline{C}_\alpha$ contains at most a fraction of α of all scenarios in $\mathbb{M}$, $\overline{C}_\alpha$ contains at least a fraction of α of all scenarios in $\mathbb{M}$.*
2. *It holds $\underline{C}_\alpha \subseteq \overline{C}_\alpha$.*

Proof.

1. It holds

$$\begin{aligned}
\Pr(\varepsilon \in \underline{C}_\alpha) &\leq \Pr(\overline{\alpha}(V(\varepsilon)) \leq \alpha) && \text{by construction of } \underline{C}_\alpha\\
&\leq \Pr(F(V(\varepsilon)) \leq \alpha) && \text{as } \overline{\alpha} \geq F\\
&= \alpha,\\
\Pr(\varepsilon \in \overline{C}_\alpha) &\geq \Pr(\underline{\alpha}(V(\varepsilon)) \leq \alpha) && \text{analogously}\\
&\geq \Pr(F(V(\varepsilon)) \leq \alpha) = \alpha && \text{as } \underline{\alpha} \leq F\\
&= \alpha.
\end{aligned}$$

2. This follows from the fact that $\underline{\alpha} \leq \overline{\alpha}$, $\underline{\alpha}$, $\overline{\alpha}$ monotone, continuous. We just regard the non trivial case that solutions $\{V_{\alpha 1}\} \subseteq V(\mathbb{M})$ of $\underline{\alpha}(V_{\alpha 1}) = \alpha$ and solutions $\{V_{\alpha 2}\} \subseteq V(\mathbb{M})$ of $\overline{\alpha}(V_{\alpha 2}) = \alpha$ exist. Then $\overline{V}_\alpha = \max\{V_{\alpha 1}\}$, $\underline{V}_\alpha = \min\{V_{\alpha 2}\}$ by definition. We show that $\underline{V}_\alpha \leq \overline{V}_\alpha$, then the proposition follows directly from the definition of $\underline{C}_\alpha$ and $\overline{C}_\alpha$. Assume that $\underline{V}_\alpha > \overline{V}_\alpha$. Then $\overline{\alpha}(\overline{V}_\alpha) \leq \overline{\alpha}(\underline{V}_\alpha)$ as $\overline{\alpha}$ is monotone. The case that $\overline{\alpha}(\overline{V}_\alpha) < \overline{\alpha}(\underline{V}_\alpha)$ contradicts to $\underline{\alpha} \leq \overline{\alpha}$ as $\overline{\alpha}(\overline{V}_\alpha) < \overline{\alpha}(\underline{V}_\alpha) = \alpha = \underline{\alpha}(\overline{V}_\alpha)$. The case that $\overline{\alpha}(\overline{V}_\alpha) = \overline{\alpha}(\underline{V}_\alpha) = \alpha$ contradicts to the definition of $\underline{V}_\alpha$ as the minimum of all solutions $\{V_{\alpha 2}\} \subseteq V(\mathbb{M})$ of $\overline{\alpha}(V_{\alpha 2}) = \alpha$.

□

In terms of robust design the regions $\underline{C}_\alpha$ and $\overline{C}_\alpha$, yield safety constraints, as a design is called **safe** if all scenarios $\varepsilon \in \overline{C}_\alpha$ satisfy the design requirements, and it is called **unsafe** if one scenario $\varepsilon \in \underline{C}_\alpha$ fails to satisfy these requirements. Between $\underline{C}_\alpha$ and $\overline{C}_\alpha$ there is a zone of **borderline** cases, where the statistical information is too scarce to fully analyze the safety. In robust design optimization, cf. Chapter 4 and Chapter 5, one looks for the optimal safe design, or for the optimal design which is not unsafe. The latter approach has the advantage that $\underline{C}_\alpha$ is always bounded and we will additionally construct it to be computationally attractive.

Let us summarize what is needed to generate a potential cloud: a potential function V has to be chosen, then appropriate bounds on the CDF F of $V(\mathbb{M})$ must be found. How to find these bounds will be described in the following Section 3.3. But how to choose the potential function? – it should be centered around the mode and reflect the 'shape' of ensembles of realizations. There are endless possibilities to make the choice. Two special cases for choices of the potential function are shown in Figure 3.1:

$$V(x) := \max_k \frac{|x^k - \mu^k|}{r^k}, \tag{3.8}$$

where $x, \mu, r \in \mathbb{R}^n$, x^k, μ^k, r^k are the k^{th} components of the vectors, defines a **box-shaped potential**.

$$V(x) := \|Ax - b\|_2^2, \tag{3.9}$$

where $x, b \in \mathbb{R}^n, A \in \mathbb{R}^{n \times n}$, defines an **ellipsoid-shaped potential**.

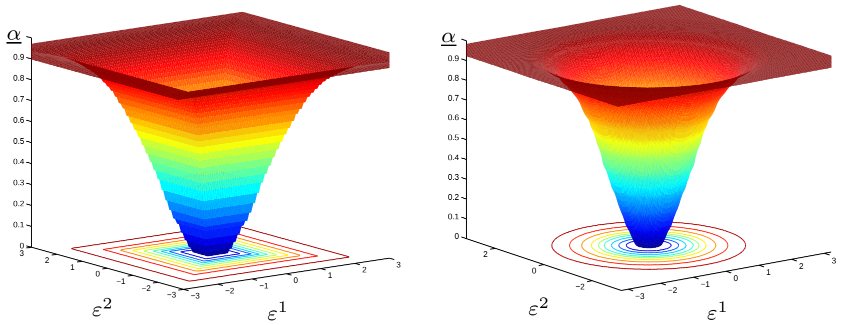


Figure 3.1: Lower level $\underline{\alpha}$ for a 2-dimensional potential cloud based on a box and ellipsoidal potential, respectively.

A variation of the shape of the potential to improve the uncertainty model will be considered in Section 3.4.

3.3 Generation of potential clouds

This section will investigate how to find appropriate bounds on the CDF F of the potential $V(\varepsilon)$. As we do not have the knowledge of F we have to approximate it before we can assign bounds on it. To this end we will make use of the well-known KS statistics [82] as suggested in the last section. That means we approximate F by an **empirical distribution** $\widetilde{F}$. The generation of an empirical distribution requires the existence of a sample S representing our uncertainties.

It depends on the given uncertainty information whether a sample already exists. We assume that *a priori* uncertainty information consists of given samples, boxes or invertible 1-dimensional marginal CDFs F_i, $i \in I \subseteq \{1, 2, \ldots, n\}$, on the n-dimensional random vector ε, without any formal knowledge about correlations or joint distributions. In case there is no sample provided or the given sample is very small, a sample has to be generated. For these cases we first use a **Latin hypercube sampling** (LHS, cf. Section 2.1.6) inspired method to generate S. This introduces some preference for a simple structure. The effect of this preference will be diminished by weighting of the sample points, cf. (3.15).

To generate N_S sample points we start with the creation of a $N_S \times \cdots \times N_S = N_S^n$ grid. In case of a given interval on ε^i the marginal grid points are chosen equidistantly in the interval. In case of a given F_i, $i \in I$, the marginal grid is transformed with respect to the marginal CDF F_i to ensure that each grid interval has the same marginal probability. Let $\alpha_S \in (0, 1]$, a confidence level for the sample generation, $p_S = 1 - \sqrt[n]{\alpha_S}$,

$$t_1 = \frac{p_S}{2},\ t_2 = t_1 + 1 \cdot \frac{1 - p_S}{N_S - 1},\ t_3 = t_1 + 2 \cdot \frac{1 - p_S}{N_S - 1}, \ldots,$$

$$t_{N_S} = t_1 + (N_S - 1) \cdot \frac{1 - p_S}{N_S - 1} = 1 - \frac{p_S}{2}, \tag{3.10}$$

then the marginal grid points are chosen as

$$g_{i1} = F_i^{-1}(t_1),\ g_{i2} = F_i^{-1}(t_2), \ldots, g_{iN_S} = F_i^{-1}(t_{N_S}). \tag{3.11}$$

3.3.1 Remark. For $i \in I$ the intervals between adjacent marginal grid points have the same marginal probability since

$$\Pr(x \in [g_{ik}, g_{ik+1}]) = F_i(g_{ik+1}) - F_i(g_{ik}) = t_{k+1} - t_k = \frac{1-p_S}{N_S - 1},$$

which is constant and independent from k, $k \in \{1, 2, \ldots, N_S - 1\}$.

From this grid the sample points $x_1, x_2, \ldots, x_{N_S}$ are chosen to satisfy the LHS condition (2.34) avoiding the grid diagonal, i.e., to avoid a choice $x_i = (g_{1i}, g_{2i}, \ldots, g_{ni})$ for $i = 1, \ldots, N_S$.

3.3.2 Remark. Assume that $I = \{1, 2, \ldots, n\}$, i.e., a marginal CDF is given for each coordinate of ε. Assume that ε^i, $i \in I$, are independent, then α_S is a confidence level for ε.

Proof. $\Pr(\varepsilon \in [g_{11}, g_{1N_S}] \times [g_{21}, g_{2N_S}] \times \cdots \times [g_{n1}, g_{nN_S}])$

$$\begin{aligned} &= \prod_{i \in I} \Pr(\varepsilon^i \in [g_i1, g_{iN_S}]) \\ &= \prod_{i=1}^{n} (F_i(g_{iN_S}) - F_i(g_{i1})) \\ &= \prod_{i=1}^{n} (t_{N_S} - t_1) \\ &= (1 - p_S)^n = \alpha_S. \end{aligned}$$

□

Our sample generation leads to the following result.

3.3.3 Proposition. *The **marginal empirical distribution***

$$\widetilde{F}_i(\xi) = \sum_{\{j \mid x_j^i \leq \xi\}} \frac{1}{N_S}, \; i \in I, \tag{3.12}$$

of our sample approximates F_i uniformly for $N_S \to \infty$ and $\alpha_S \to 1$.

Proof. For $\alpha_S \to 1$ this proposition corresponds to the Glivenko-Cantelli theorem (cf., e.g., [22]) by construction of our sample points. It states, that

$$\lim_{N_S\to\infty}\sup_{\xi}|\widetilde{F}_i(\xi)-F_i(\xi)|=0$$

almost surely. □

3.3.4 Remark. The confidence level α_S can be freely chosen from $(0,1]$. A choice of $\alpha_S=1$ allows $g_{i1}=-\infty$, $g_{iN_S}=\infty$. In practice one chooses α_S smaller than but close to 1. We use $\alpha_S=0.998$ and $N_S=1000$ in our applications. Also note that by the use of $\alpha_S<1$ the sample generation can distinguish between mere interval information and a marginal uniform distribution: the information that $\varepsilon^i\in[a,b]$ will result in a different sample than the information that ε^i is uniformly distributed in $[a,b]$.

By Proposition 3.3.3 the generated sample $S:=\{x_1,x_2,\dots,x_{N_S}\}$ represents the marginal CDFs arbitrarily well. However after a modification of S, e.g., by cutting off sample points as we will do later providing *a posteriori* information, the marginal empirical distributions will no longer approximate the marginal CDFs, and thus the *a priori* information is no longer respected. An assignment of **weights** to the sample points is necessary to preserve the marginal CDFs. In order to do so the weights $w_1,w_2,\dots,w_{N_S}\in[0,1]$, corresponding to the sample points $x_1,x_2,\dots,x_{N_S}$, are required to satisfy the following conditions:

Let π_j be a sorting permutation of $\{1,2,\dots,N_S\}$, such that $x^j_{\pi_k(1)}\le x^j_{\pi_k(2)}\le\dots\le x^j_{\pi_k(N_S)}$. Let again I be the index set of those entries of the random vector ε where a marginal CDF $F_i, i\in I$ is given. Then the weights should satisfy for all $i\in I,k\in\{1,2,\dots,N_S\}$

$$\sum_{j=1}^{k}w_{\pi_i(j)}\in[F_i(x^i_{\pi_i(k)})-d,F_i(x^i_{\pi_i(k)})+d],\ \sum_{\ell=1}^{N_S}w_\ell=1. \tag{3.13}$$

The function

$$\widetilde{F}_i(\xi):=\sum_{\{j|x^i_j\le\xi\}}w_j \tag{3.14}$$

is a **weighted marginal empirical distribution**. For trivial weights, $w_1 = w_2 = \cdots = w_{N_S} = \frac{1}{N_S}$, $\widetilde{F}_i$ is a standard empirical distribution as in Proposition 3.3.3.

To achieve weights satisfying (3.13) we formulate the following linear program,

$$\begin{aligned} \min_{w_1,w_2,\dots,w_{N_S}} \quad & d \\ \text{s.t.} \quad & \widetilde{F}_i(\xi_k^i) \in [F_i(\xi_k^i) - d, F_i(\xi_k^i) + d] \text{ for all } i,\ \xi_k^i \\ & \sum_{i=1}^{N_S} w_i = 1 \\ & w_i \geq 0 \text{ for all } i \\ & w_i \leq \frac{10}{N_S} \text{ for all } i \\ & d \geq 0 \end{aligned} \tag{3.15}$$

where ξ_k^i are some given interpolation points on the related margin, $\widetilde{F}_i$ the weighted empirical marginal distributions as defined in (3.14). The number of interpolation points allows to adjust the computational effort, cf. Section 3.5. We constrain the maximum of the weights to be less than 10 times the mean of all weights, i.e., $\frac{1}{N_S}$, to avoid extraordinarily large weights, which may otherwise occur in higher dimensions.

The smallest d for which (3.13) holds is then

$$d_{\min} = \max_{i \in I, j=1,2,\dots,N_S} |\widetilde{F}_i(x_j^i) - F_i(x_j^i)| \tag{3.16}$$

the maximum deviation of $\widetilde{F}_i$ from F_i, $i \in I$. By the weight computation we get a **weighted empirical distribution**

$$\widetilde{F}(\xi) := \sum_{\{j \mid V(x_j) \leq \xi\}} w_j \tag{3.17}$$

approximating the CDF of $V(\varepsilon)$. The achievement of weights satisfying (3.13) means, that all uncertainty information of the marginal CDFs is reflected in the construction of $\widetilde{F}$, even if an *a posterior* modification of S has excluded some sample points. The information given as boxes or sample data is reflected anyway as this does not imply additional constraints to (3.13).

The constraints (3.13) require the weights to represent the marginal CDFs with some reasonable margin d. In other words, the weighted marginal empirical distributions $\widetilde{F}_i, i \in I$ should not differ from the given marginal CDF F_i by more than d. In practice, one chooses $d = d_{\text{KS},1}$ with a version of the KS statistics [82] as introduced in Section 2.1.3. We work with the formula from Press et al. [131],

$$d_{\text{KS},1} = \frac{\phi^{-1}(a)}{\sqrt{N_S} + 0.12 + \frac{0.11}{\sqrt{N_S}}}, \tag{3.18}$$

where ϕ is the Kolmogorov function $\phi(\lambda) = \sum_{k=-\infty}^{+\infty}(-1)^k e^{-2k^2\lambda^2}$, cf. (2.27), and $a = \alpha_w$ the confidence in the KS theorem.

3.3.5 Remark. The choice of $d_{\text{KS},1}$ is reasonable in practice although not precisely fitting the assumptions used for the KS theorem. As mentioned in Section 2.1.3 we invert the statement of the KS test to compute $d_{\text{KS},1}$. However, for the KS test the empirical distribution $\widetilde{F}_i$ is assumed to be independent from the choice of the compared distribution F_i. In our case $\widetilde{F}_i$ and F_i are not chosen independently, as the sample which determines $\widetilde{F}_i$ is generated from information about F_i.

If $d_{\min} > d_{\text{KS},1}$ weights satisfying (3.13) can only be achieved with $d > d_{\text{KS},1}$, the relaxation $d_{\min}$ gives us a soft indicator for the quality of the approximation which will be useful to construct bounds on the CDF F of $V(\varepsilon)$. Remember that after the approximation of F with $\widetilde{F}$, we are just one step away from generating a potential cloud. The last step is seeking an appropriate bound for F.

From the knowledge of $d_{\text{KS},1}$, and $d_{\min}$ we compute $d_{\text{KS},2}$ similar to before (3.18), for a fixed confidence $a = \alpha_b$:

$$d_{\text{KS},2} = \frac{\phi^{-1}(a)}{\sqrt{N_S} + 0.12 + \frac{0.11}{\sqrt{N_S}}} \cdot \max(1, \frac{d_{\min}}{d_{\text{KS},1}}), \tag{3.19}$$

with the approximation quality factor $\frac{d_{\min}}{d_{\text{KS},1}}$. Now we define $\overline{F} := \min(\widetilde{F} +$

$d_{\mathrm{KS},2}, 1)$ and $\underline{F} := \max(\widetilde{F} - d_{\mathrm{KS},2}, 0)$ and fit these two step functions to smooth, monotone lower bounds $\underline{\alpha}$ and upper bounds $\overline{\alpha}$, cf. Figure 3.2. Observe that if the quality of our approximation with $\widetilde{F}$ or the sample size N_S is decreased, the width of the bounds is increased correspondingly.

Thus we have found an appropriate empirical bound of the CDF of $V(\varepsilon)$ and according to Theorem 3.2.1 we have generated a potential cloud that fulfills the conditions that define a cloud via the mapping $\chi : x \to [\underline{\alpha}(V(x)), \overline{\alpha}(V(x))]$.

Note that if a weighted sample of real statistical data is actually available (in the simplest case all weights are $\frac{1}{N_S}$), we can directly compute $\underline{F}$ and $\overline{F}$ from $\widetilde{F}$ setting $d = d_{\mathrm{KS},1}$.

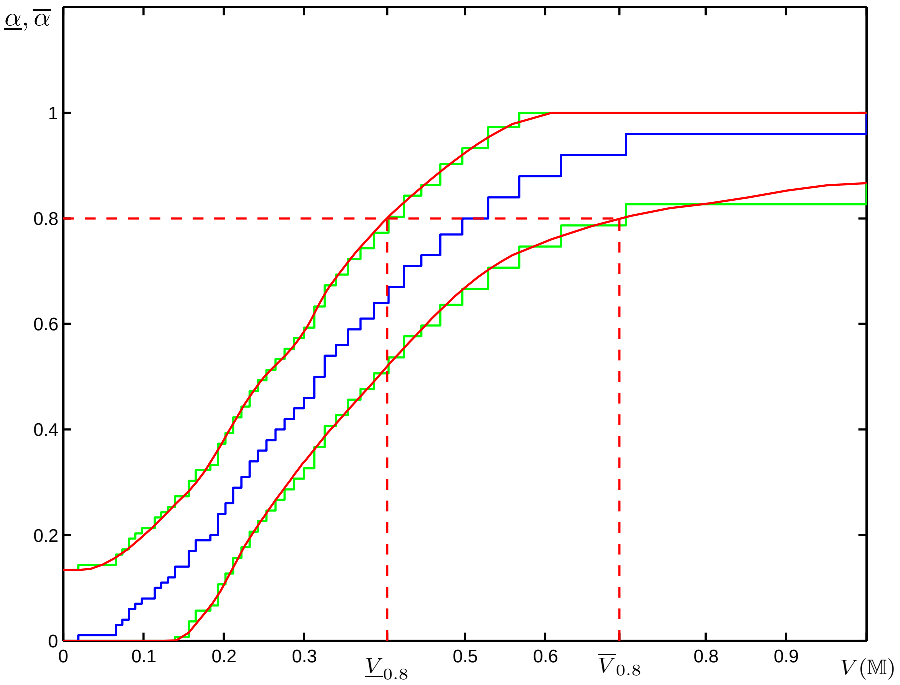


Figure 3.2: The smoothed lower bounds $\underline{\alpha}$ and upper bounds $\overline{\alpha}$ enclosing the empirical distribution of $V(\varepsilon)$. The mapping $x \to [\underline{\alpha}(V(x)), \overline{\alpha}(V(x))]$ is a potential cloud (cf. Section 3.2). The horizontal cut $\alpha = 0.8$ is illustrated and the corresponding $\underline{V}_\alpha$ and $\overline{V}_\alpha$, given by $\overline{\alpha}(\underline{V}_\alpha) = \alpha$ and $\underline{\alpha}(\overline{V}_\alpha) = \alpha$, respectively.

The cloud represents the given information and now enables us to interpret the potential level maps as confidence regions $\{x \mid V(x) \leq V_\alpha\}$ for the random vector ε: the **worst-case relevant** region is defined as $\underline{C}_\alpha$ (cf. Section 3.2), which is not empty if $\alpha \geq d_{\mathrm{KS},2}$ which is typically the case (unless α is very low and thus of little interest as a confidence level).

Thus clouds give an intuition and guideline how to construct confidence regions for safety constraints. To this end we have combined several different theoretical means: potential functions, KS statistics to approximate CDFs with empirical distributions and estimate bounds, sample generation methods, and weighting techniques.

3.4 The choice of the potential

The choice of the potential function V that determines the shape of the corresponding potential cloud can be chosen freely before cloud generation. In Section 3.2 we already introduced the special cases of box-shaped (3.8) and ellipsoid-shaped (3.9) potential choices. Now we will investigate some further choices of V and reflect on what characterizes a **good** choice of V.

The quality of the choice of V can be assessed by looking at the shape of the confidence regions $\underline{C}_\alpha$ of the corresponding cloud. Figure 3.3 and Figure 3.4 visualize two confidence regions for the same confidence level, but different potentials V, in two different cases for the distribution of ε. Figure 3.3 shows that uncertainties can be described reasonably by different clouds. Figure 3.4 illustrates that a shape of V which reflects the shape of the point set representative for the uncertainties often means a less pessimistic worst-case analysis with the cloud and can thus indicate a good choice of V. We emphasize that a poor choice of the potential makes the worst-case analysis more pessimistic as the confidence regions are larger, but will still result in a valid robust uncertainty handling.

We are looking for a way to find a good choice of V that is computationally attractive and giving the possibility to improve the potential iteratively. Such a choice should allow for a simple computational realization of the confidence regions, e.g., by linear constraints. This leads us to the investigation of **polyhedron-shaped potentials** as a generalization of box-shaped poten-

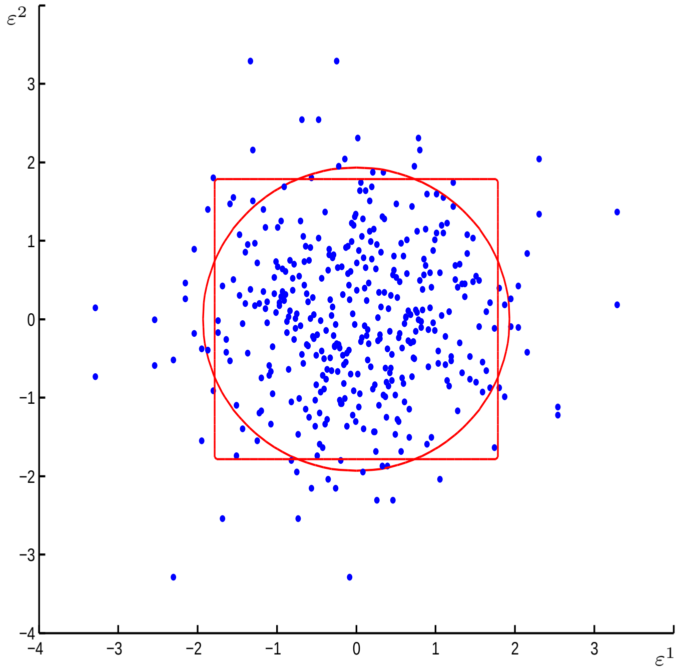


Figure 3.3: The regions $\underline{C}_{0.95}$ for two different choices of V, box- and circle-shaped, respectively. The 2-dimensional sample belongs to two independent $N(0,1)$-distributed random variables ε^1 and ε^2. The two choices of V both describe the shape of the point set reasonably.

tials. A polyhedron potential centered at $m \in \mathbb{R}^n$ can be defined as :

$$V_p(x) := \max_k \frac{(A(x-m))^k}{b^k}, \tag{3.20}$$

where $(A(x-m))^k, b^k$ the k^{th} component of the vectors $A(x-m)$ and b, respectively.

But how to achieve a polyhedron that reflects the given uncertainty information in the best way? As mentioned previously we assume the initially available uncertainty information to consist of given samples, boxes or marginal distributions. Furthermore we assume that additional information can be provided by an expert as unformalized **dependency constraints**.

After generation of a sample S as described in Section 3.3 we define a box

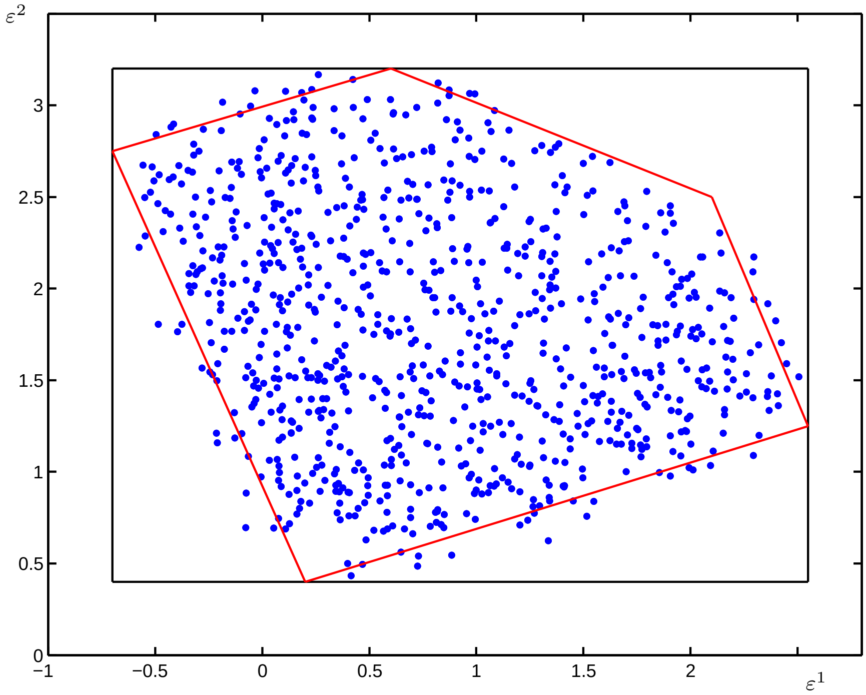


Figure 3.4: The regions $\overline{C}_\alpha$ for a box-shaped and polyhedral-shaped V. In this case a polyhedral shape of V should be preferred as it is a better representation of the given uncertain scenarios and will lead to a less pessimistic worst-case analysis.

b_0 containing 100% of the sample points by

$$b_0 := [g_{11}, g_{1N_S}] \times [g_{21}, g_{2N_S}] \times \cdots \times [g_{n1}, g_{nN_S}], \tag{3.21}$$

and we define our potential V_0 box-shaped as in (3.8) taking the value 1 on the margin of b_0, i.e.,

$$\mu^k = \frac{g_{k1} + g_{kN_S}}{2}, \ r^k = \frac{g_{kN_S} - g_{k1}}{2}. \tag{3.22}$$

Based on expert knowledge, a user-defined variation of V_0 can be performed afterwards by cutting off sample points deemed irrelevant for the worst-case: The optimization phase, cf. Chapter 5, provides a worst-case scenario which is highlighted in a **graphical user interface** (GUI). The expert can decide to exclude, e.g., the worst-case or different scenarios, based on his technical knowledge. Assume that the linear constraints $A(x - \mu) \leq b$ represent the exclusion of sample points and the box constraint from b_0, we define our polyhedron shaped potential as in (3.20) with $m = \mu$. The exclusion in the GUI will be conducted in 1-dimensional or 2-dimensional projections of ε as we will see in Chapter 6, cf. Figure 6.2, thus the matrix A becomes sparse. By means of this exclusion an expert can specify the uncertainty information in the form of dependency constraints adaptively, even if the expert knowledge is only little formalized, resulting in a polyhedron shaped potential.

3.4.1 Remark. There is a close relationship between the dependency constraints and correlation information. Consider the 2-dimensional normally distributed case centered at 0, with non-zero correlation. Then the confidence regions are ellipse shaped $||Ax||_2^2 \leq$ const. If one approximates the ellipse with a parallelepiped $||Ax||_\infty \leq$ const you end up in a polyhedron shaped confidence region, see, e.g., Figure 3.5.

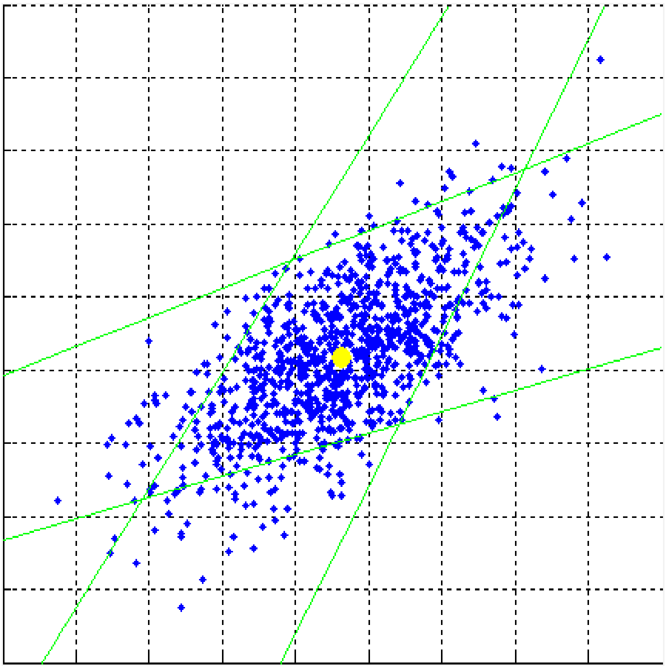


Figure 3.5: Sample points in a 2-dimensional normally distributed example with non-zero correlation. The elliptical confidence regions can be reasonably approximated by linear constraints.

This potential, originating from a box potential, is suitable for symmetric samples, but: If some uncertain variables are described by asymmetric marginal probability densities, a better choice V_t of the potential could be achieved by an appropriate **coordinate transformation** $\mathcal{T}$, i.e.,

$$V_{\mathcal{T}}(x) := V_p(\mathcal{T}(x)). \tag{3.23}$$

An appropriate transformation would be, e.g., a logarithmic transformation of ε^i if $F_i : \mathbb{R}^+ \to [0,1]$. An example of a 2-dimensional potential cloud with $V = V_{\mathcal{T}}$ is visualized in Figure 3.6.

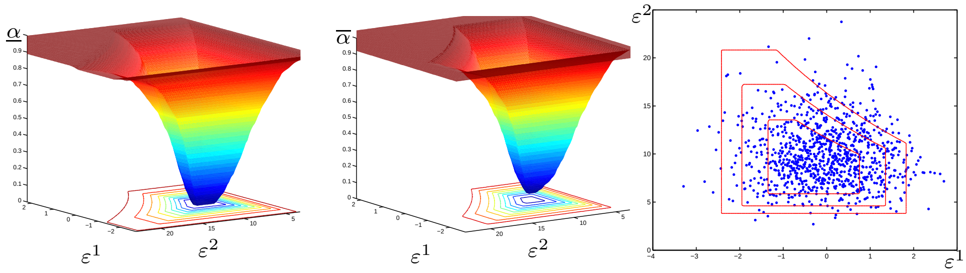


Figure 3.6: On the left, a 2-dimensional potential cloud mapping, i.e., the maps $x \to \underline{\alpha}(V(x))$ and $x \to \overline{\alpha}(V(x))$; on the right, the contour lines of $\underline{\alpha} \circ V$. The marginal distributions for $\varepsilon = (\varepsilon^1, \varepsilon^2)$ are a $N(0,1)$ and a $\Gamma(10,1)$ distribution, respectively.

We can observe the advantage of a transformed potential in Figure 3.7. Without transformation the functions $\underline{\alpha}$ and $\overline{\alpha}$ are obviously steeper, and for α close to 1 the solution V_α of $\overline{\alpha}(V_\alpha) = \alpha$ is much closer to 1 than in the transformed case, which leads to larger confidence regions, mostly implicating a more pessimistic worst-case analysis. The reason for that becomes apparent looking at Figure 3.8. The confidence regions for the transformed box potential are obviously smaller than for the non-transformed potential.

3.5 Computational ways to generate potential clouds

Now we give a survey on the algorithms that realize the described methods computationally. We will directly follow the considerations from the previous sections. We will give code examples for the grid selection of LHS (2.34), for the weight computation (3.15) and for the smooth fit of the bounds $\underline{\alpha}$, $\overline{\alpha}$.

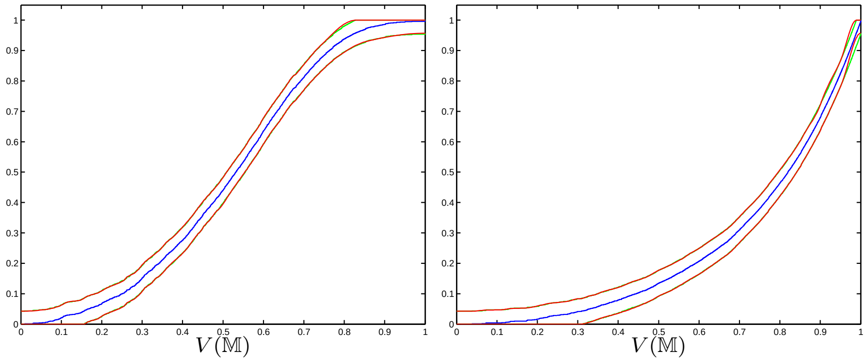


Figure 3.7: The lower bounds $\underline{\alpha}$ and upper bounds $\overline{\alpha}$ for a potential cloud with transformation (left figure) and without transformation (right figure). The marginal distributions for $\varepsilon = (\varepsilon^1, \varepsilon^2)$ are a $N(0,1)$ and a $\Gamma(10,1)$ distribution, respectively.

The first computation occurring in our approach is the sample generation based on LHS. The algorithm for the sample generation is a straightforward realization of the theoretical description in Section 2.1.6. After construction of a grid as described, with a matrix `grid` of size $N_s \times n$ containing the marginal grid points, we generate the sample points in a $N_s \times n$ matrix `samplepoints`:

```
per=zeros(N,n); % initialization of per
for i=1:n
    while 1
        per(:,i)=randperm(N);
        b=1;
        % this construction avoids a sample
        % on the grid diagonal:
        for j=1:i-1
            if ~all(per(:,i)==per(:,j))
                b=1;
                break;
            else
                b=0;
            end;
        end;
        if b==1
            break;
        end;
    end;
end;
```

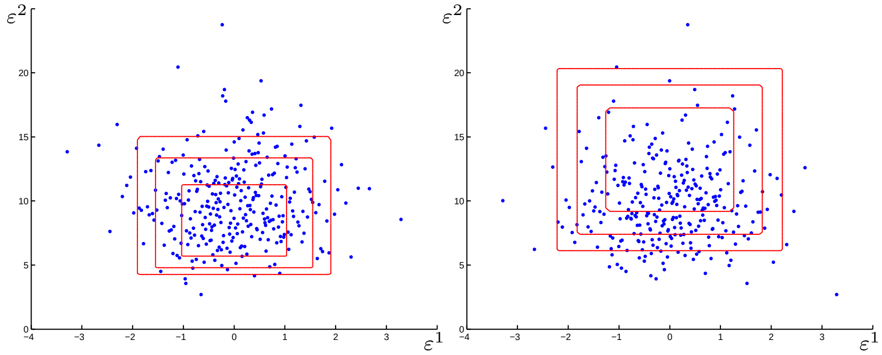


Figure 3.8: The figures show the regions $\underline{C}_\alpha$, $\alpha = 0.5, 0.8, 0.95$ for a box potential cloud with (left figure) and without (right figure) transformation. The two samples are generated for ε distributed as in Figure 3.7.

```
for i=1:N
    for j=1:n
        samplepoints(i,j)=grid(per(i,j),j);
    end;
end;
```

Recall the notation used for the weight computation. Let `dind` correspond to I, `w` to w, `d` to $d_{\mathrm{KS},1}$, `br` be the matrix containing interpolation points on the margin, `nbr` the number of interpolation points which is chosen constant on each margin, `pdf` an array of strings looking like, e.g., `'normcdf(x,0,1)'`. Then we can realize (3.15) in Matlab as follows:

```
for i=1:n
    x=br(:,i);       % project interpolation points to the related margin
    cdf(:,i)=eval(pdf{i});   % CDF values of the interpolation points
    for j=1:nbr
        % permutation to construct the marginal empirical distributions:
        perb{j,i}=find(sample(:,i)<=br(j,i));
    end;
end;

cvx_begin
variable w(N);
variable e;
minimize(e);
subject to
e>0;
```

```
sum(w)==1;
w>=zeros(size(w));
w<=(10/N).*ones(size(w));
for i=1:length(dind)
    for j=1:nbr
        sum(w(perb{j,dind(i)}))>=cdf(j,dind(i))-e*d;
        sum(w(perb{j,dind(i)}))<=cdf(j,dind(i))+e*d;
    end;
end;
cvx_end
```

The problem is solved using the Matlab convex programming toolbox CVX developed by Grant & Boyd [63]. It should be remarked that we make use of the Matlab Statistics Toolbox to evaluate probability distributions. Also note that we use the Matlab `eval` function for simplicity of the code. For performance reasons one should use function pointers instead.

After the weight computation we have all $\widetilde{F}_i$ and $\widetilde{F}$ as defined in equation (3.14) and (3.17). We can compute $d_{\min}$, the maximum deviation of $\widetilde{F}_i$ from F_i, $i \in I$, as in (3.16).

The number of constraints in the linear program (3.15) is $N_{\text{CVX}} = 2N_\xi|I| + 2N_S + 2$, where N_ξ is the number of interpolation points on each margin, $|I|$ the cardinality of I. A reasonable value for N_{CVX} with regard to the computation time is between 2000 and 8000 on an Intel Core 2 Laptop CPU T5500 1.66 GHz computer with 2 GB RAM. Observe that in higher dimensions one may have to decrease the number of interpolation points, and thus the weights are likely to satisfy the constraints (3.13) only for $d_{\min} > d_{\text{KS},1}$. This is the point where the curse of dimensionality can appear during the computations if $d_{\min} \gg d_{\text{KS},1}$.

Now we have achieved $\widetilde{F}$ and $d_{\text{KS},2}$ from (3.19) and the next step is to fit $\overline{F} := \min(\widetilde{F} + d_{\text{KS},2}, 1)$ and $\underline{F} := \max(\widetilde{F} - d_{\text{KS},2}, 0)$ smoothly. We first fit these step functions with a continuous piecewise linear function `lineval`, and afterwards make use of piecewise cubic Hermite splines, cf. Fritsch & Carlson [57], for the final smooth fit via `pchip`. The main characteristics of these splines for our purposes is the fact that they respect monotonicity.

We start within the interval `[x0,x1]` where the a priori continuous piecewise linear fit, that is `lineval`, is non constant. Let `x` be a vector of initial interpolation points ∈(x0,x1), lineval(x) be the evaluation of lineval at x. Let v∈(x0,x1) a set of points where the spline fit is evaluated to be compared with lineval, h some tolerance level to serve as stopping criterion for the spline fit.

```
y=lineval(x);
lx=length(x);
for i=1:ceil(sqrt(N))-lx-1  % max. of sqrt(N) interpolation points
    s=pchip(x,y,v);    % Hermite spline
    % check stopping criterion
    if all(abs(lineval(v)-s)<=h)
        break;
    else
        % find index of v with maximal deviation
        [a,ind]=max(abs(lineval(v)-s));
        % create a new interpolation point there
        xnew=v(ind);
        ynew=lineval(xnew);
        x=[x,xnew];
        y=[y,ynew];
        [x,p]=sort(x);
        y=y(p);
    end;
end;
```

Eventually x ∈(x0,x1) are the interpolation points for the spline fit. We add two more points, namely x0 and x1, where the slope of the spline is forced to take value 0, this is possible for Hermite splines. With this final set of interpolation points we achieve a smooth fit on the whole range of V, cf., e.g., Figure 3.2.

The fit of the smooth bounds completes the computational cloud generation with respect to Section 3.3.

For the algorithms described above we initially choose V_0 as the potential function. But due to Section 3.4 we have implemented an adaptive variation of the potential to a polyhedron shape via a graphical user interface which will be presented in detail in Chapter 6.

Chapter 4

Design optimization

A classic approach to design optimization, without taking uncertainties into account, leads to decision support for engineers, or even autonomous design, but to a design which completely lacks robustness. We want to safeguard the design against worst-case scenarios, i.e., the design should not only satisfy the given requirements on functionalities, but should also work under uncertain, adverse conditions that may show up during employment of the designed object. This will involve methods for uncertainty modeling in the optimization phase.

In this chapter we will elaborate a solution approach to design optimization without taking uncertainty into account. Thus we prepare for the next chapter, which finally combines uncertainty handling (see the last chapter) and design optimization (see this chapter).

We start with a formal statement of the optimization problem in Section 4.1. Afterwards, in Section 4.2, we point out the difficulties related with the problem. In Section 4.3 heuristic approaches to the solution of the problem are presented. An implementation can be found in Section 4.4.

4.1 Problem formulation

In an underlying model of a given structure, e.g., a spacecraft component, with several inputs and outputs, we denote by x the vector containing all **output variables**, and by z the vector containing all **input variables**. The inputs contained in z can be divided into **global input variables** u and **design variables** v.

The design variables are determined by the so-called design **choice variables**. A choice variable is a univariate variable controllable for the design. The choice variables can be continuous, e.g., the diameter of an antenna, or discrete, e.g., the choice of a thruster from a set of different thruster types. Let θ be the vector of design choice variables $\theta^1, \theta^2, \ldots, \theta^{n_o}$. Let I_d be the index set of choice variables which are discrete and I_c be the index set of choice variables which are continuous, $I_d \cup I_c = \{1, 2, \ldots, n_o\}$, $I_d \cap I_c = \emptyset$.

In the discrete case, $i \in I_d$, the choice variable θ^i determines the value of n_i design variables. For example, if θ^i was the choice of a thruster, each choice could be specified by the thrust and specific impulse of the thruster. Thrust and specific impulse would be design variables v_1^i and v_2^i, and $n_i = 2$ in this example. Let $1, 2, \ldots, N_i$ be the possible choices for θ^i, $i \in I_d$, then the discrete choice variable θ^i corresponds to a finite set of N_i points $(v_1^i, v_2^i, \ldots, v_{n_i}^i) \in \mathbb{R}^{n_i}$. Usually this set is provided in an $N_i \times n_i$ table $(\tau_{j,k}^i)$ (see, e.g., Table B.3, $N_i = 30$, $n_i = 3$), and define

$$Z^i(\theta^i) := (\tau^i_{\theta^i,1}, \tau^i_{\theta^i,2}, \ldots, \tau^i_{\theta^i,n_i}) \text{ for } \theta^i \in \{1, 2, \ldots, N_i\}, \tag{4.1}$$

i.e., the ${\theta^i}^{\text{th}}$ row of $(\tau_{j,k}^i)$. In the continuous case, $i \in I_c$, the choice variable θ^i can be regarded as a design variable controllable in a given interval $[\underline{\theta^i}, \overline{\theta^i}]$, i.e.,

$$Z^i(\theta^i) := \theta^i \text{ for } \theta^i \in [\underline{\theta^i}, \overline{\theta^i}]. \tag{4.2}$$

A global input variable is an external input with a nominal value that cannot be controlled for the underlying model, this could be, e.g., a specific temperature.

The complete vector of inputs z has the length $\sum_{i \in I_d} n_i + |\{i \in I_c\}| +$

length(u), where u is the vector of global inputs at their nominal values. Let

$$Z(\theta) := (u, Z^1(\theta^1), Z^2(\theta^2), \dots, Z^{n_o}(\theta^{n_o})). \tag{4.3}$$

We call Z a table mapping as the nontrivial parts of Z consist of the tables $(\tau^i_{j,k})$. The mapping Z assigns an input vector z to a given design choice θ.

Both design and global input variables can be uncertain. For the classic problem formulation of design optimization we consider these variables to have the nominal value and do not take any uncertainties into account. We assume that the design optimization problem can be formulated as follows:

$$\begin{aligned} \min_{\theta} \quad & g(x) & \text{(objective function)} \\ \text{s.t.} \quad & z = Z(\theta) & \text{(table constraints)} \\ & G(x, z) = 0 & \text{(functional constraints)} \\ & \theta \in T & \text{(selection constraints)} \end{aligned} \tag{4.4}$$

where the design objective $g(x)$ is a function of the output variables of the underlying model. The **table constraints** assign to each choice θ a vector z of input variables. The **functional constraints** express the functional relationships defined in the underlying model. It is assumed that the number of equations and the number of output variables is the same (i.e., $\dim G = \dim x$), and that the equations are (at least locally) uniquely solvable for x. The **selection constraints** specify which choices are allowed for each choice variable, i.e.,

$$\theta^i \in \begin{cases} \{1, 2, \dots, N_i\} & \text{if } i \in I_d, \\ [\underline{\theta^i}, \overline{\theta^i}] & \text{if } i \in I_c. \end{cases} \tag{4.5}$$

4.2 Difficulties

The problem formulated in the last section features several difficulties of **most complex nature**. The variable types can be both continuous and integer, so the problem comes as a **mixed integer nonlinear programming** problem (MINLP). MINLP can, e.g., entail combinatorial explosion, it is still a recent research direction which has not yet matured, but we will not go very much into the details of it in this study. Profound difficulties arise from the fact that the functional constraints, represented by G, can have strong nonlinearities and can contain branching decisions such as case differentiation (implemented as, e.g., if-structures in the code) which leads to discontinuities.

Common techniques to find local or global solutions of problem (4.4) can be separated into **symbolic** and merely **heuristic** approaches. Symbolic solvers require the underlying model to be formulated in a symbolic solver language like Ampl (cf. Fourer et al. [56]), Gams (cf. Brooke et al. [18]), or Lingo (cf. Schrage [141]). Thus they can, e.g., compute derivatives making use of methods like automatic differentiation.

The formulation of the underlying model in a symbolic solver language can become a very difficult task facing real-life problems since the people who implement the underlying model for the design are sometimes working independently from the optimization phase. A semi-automatic translation of an underlying Matlab model can be found in Neumaier et al. [118]. This study also examines the capabilities of symbolic solvers with regard to the difficulties mentioned. It turned out that these solvers were unable to provide solutions for the real-life applications discussed in Chapter 7.

As we assumed that the functional constraints of the underlying model can be solved numerically for x given z we can treat them as a black box function $x = G_o(z)$ and make use of specific strategies to search the space of allowed inputs $z = Z(\theta)$, $\theta \in T$. The relationship between θ and $g(x)$ is often modeled by a surrogate function. From the knowledge of the surrogate function the heuristic methods can propose guesses for optimal solutions, or they can improve the surrogate model by further evaluations of G_o. There are not only surrogate function based heuristics but also many more, e.g., evolution strategies [65], or based on clustering [27].

In view of the difficulties mentioned we decide to approach the solution of (4.4) by means of heuristics, as described in the next section.

4.3 Heuristic strategies

In this section we present a heuristic strategy to tackle the optimization problem (4.4) stated in Section 4.1. The method we develop is based on **separable underestimation**. It exploits the characteristics of the problem, takes advantage of the discrete nature of many of the choice variables involved in real-life design, supporting, at the same time, continuous choice variables.

The functional constraints in problem (4.4) are given by the equation $G(x,z)=0$ which we assume to be uniquely numerically solvable for x. For given z we solve the multivariate zero-finding problem $G(x,z)=0$ numerically and define $G_o(z) := x$, the solution of the zero-finding problem. Note that the functional constraints in (4.4) now can be formulated as $G_o(Z(\theta)) = x$ after inserting the table constraints $z = Z(\theta)$. Then a substitution of x in the objective function of problem (4.4) leads to

$$g(x) = g(G_o(Z(\theta))) =: G_{\mathrm{bb}}(\theta), \tag{4.6}$$

which is the relation between θ and $g(x)$ described in the last section. Thus we can incorporate the functional constraints and the objective function of problem (4.4) in the black box $G_{\mathrm{bb}}(\theta)$ and rephrase (4.4) as:

$$\begin{aligned} \min_{\theta} \;\; & G_{\mathrm{bb}}(\theta) \\ \text{s.t.} \;\; & \theta \in T \end{aligned} \tag{4.7}$$

Remember θ is the vector of design choice variables $\theta^1, \theta^2, \ldots, \theta^{n_o}$. We seek a separable underestimator $q(\theta)$ for the objective function defined by

$$q(\theta) := \sum_{i=1}^{n_o} q_i(\theta^i). \tag{4.8}$$

Assume that the black box G_{bb} has been evaluated N_o times resulting in the function evaluations $G_{\mathrm{bb}_1}, G_{\mathrm{bb}_2}, \ldots, G_{\mathrm{bb}_{N_o}}$ for the design choices $\theta_1, \ldots, \theta_{N_o} \in T$. Let $l \in \{1, 2, \ldots, N_o\}$. For a discrete choice θ_l^i, we define $q_i(\theta^i)$ simply as a constant. For a continuous choice θ_l^i, we define $q_i(\theta^i)$ by a quadratic expression with the two constants q_{i1} and q_{i2}. Thus we have

$$q_i(\theta_l^i) := \begin{cases} q_{i,\theta_l^i},\ \theta_l^i \in \{1, 2, \ldots, N_i\} & \text{if } i \in I_d, \\ q_{i1}\theta_l^i + q_{i2}{\theta_l^i}^2 & \text{if } i \in I_c. \end{cases} \tag{4.9}$$

If $I_d = \emptyset$ we add an integer choice θ^i with $N_i = 1$ artificially to represent the constant part which is left out in the definition of q_i, $i \in I_c$.

The vectors q_i of constants have the length N_i for $i \in I_d$, and 2 for $i \in I_c$. They are treated as variables q_i in an LP with constant objective function subject to the constraints

$$\sum_{i=1}^{n_o} q_i(\theta_l^i) \le G_{\mathrm{bb}_l},\ l = 1, 2, \ldots, N_o. \tag{4.10}$$

To ensure that many constraints in (4.10) will be active we solve a modified version of the above LP. We pick a subset of the design choices, i.e., θ_l, $l \in I_a$, $I_a \subseteq \{1, 2, \ldots, N_o\}$, such that

$$\begin{aligned} \min_{q_j, j\in\{1,2,\ldots,n_o\}} \quad & \sum_{i=1}^{n_o} q_i^T q_i \\ \text{s.t.} \quad & \sum_{i=1}^{n_o} q_i(\theta_l^i) = G_{\mathrm{bb}_l}, l \in I_a \end{aligned} \tag{4.11}$$

has a feasible solution. Afterwards we compute the differences $G_{\mathrm{bb}_l} - \sum_{i=1}^{n_o} q_i(\theta_l^i)$ for all $l \in \{1, 2, \ldots, N_o\}$ and find those l with the largest negative values for the difference redefining the set I_a, and those l with the largest positive values defining the set I_{ia}. Then we continue with solving

$$\begin{aligned} \min_{q_j, j\in\{1,2,\ldots,n_o\}} \quad & \sum_{i=1}^{n_o} q_i^T q_i + \sum_{l\in I_{ia}} (G_{\mathrm{bb}_l} - \sum_{k=1}^{n_o} q_k(\theta_l^k))^2 \\ \text{s.t.} \quad & \sum_{i=1}^{n_o} q_i(\theta_l^i) = G_{\mathrm{bb}_l}, l \in I_a \end{aligned} \tag{4.12}$$

and iterate this procedure until either $G_{\mathrm{bb}_l} - \sum_{i=1}^{n_o} q_i(\theta_l^i)$ is nonnegative for all $l \in \{1, 2, \ldots, N_o\}$, so that q is constructed to be an underestimator of G_{bb} at the given points $\theta_1, \theta_2, \ldots, \theta_{N_o}$ satisfying (4.10), or a maximum number of iterations is reached. In the latter case

$$q_{\mathrm{maxniter}}(\theta) := \sum_{i=1}^{n_o} q_i(\theta^i) - \max_{l\in\{1,2,\dots,N_o\}} (G_{\mathrm{bb}_l} - \sum_{i=1}^{n_o} q_i(\theta_l^i)) \tag{4.13}$$

is an underestimator for G_{bb} at the given points $\theta_1, \theta_2, \dots, \theta_{N_o}$ anyway. The underestimator $q(\theta)$ is separable and can be easily minimized via

$$\theta^i = \min_{j\in\{1,2,\dots,N_i\}} q_{i,j} \qquad \text{if } i \in I_d \tag{4.14}$$

$$\theta^i = \begin{cases} -\frac{q_{i,1}}{2q_{i,2}} & \text{if } q_{i,2} \neq 0 \\ \overline{\theta^i} & \text{if } q_{i,2} = 0,\ q_{i,1} < 0 \\ \underline{\theta^i} & \text{if } q_{i,2} = 0,\ q_{i,1} \geq 0 \end{cases} \qquad \text{if } i \in I_c. \tag{4.15}$$

In addition to the development of the method of separable underestimation we also make use of further strategies to find a solution of the optimization problem (4.7). The first one is called Snobfit, developed by Huyer & Neumaier [72]. It is surrogate function based and fits a quadratic model for G_{bb}. Integers are treated as continuous variables and rounded to a grid with step width 1. After the analysis of the quadratic model, a user-defined number of choices θ is requested to be computed from $G_{\mathrm{bb}}(\theta)$ to update the model in the next iteration. The requests also contain the minimizer for the current quadratic model. The stopping criterion can be freely chosen within the implementation of the method. For surrogate function based strategies it is reasonable to prepare the integer choices for the black box model prior to the optimization. Remember each value $1, 2, \dots, N_i$ of a discrete choice variable θ^i, $i \in I_d$ corresponds to a point in $\mathbb{R}^{n_i}$ via $Z^i(\theta^i)$. The involved table τ^i represents a sequence of N_i points in $\mathbb{R}^{n_i}$. This sequence should be sorted in a reasonable way, e.g., seeking monotonicity in each column of τ^i which correspond to the design variables, to impose a good surrogate function fit.

Another method we use is based on evolution strategy with covariance matrix adaptation, called CMAES, developed by Hansen & Ostermeier [65]. It is not surrogate function based, but a specific stochastic method to sample the search space T. This strategy requires quite a large amount of function evaluations. Integers are also treated as continuous variables rounded to the next integer value.

Finally the minimizers that result from all methods used are taken as starting points for a limited global search, i.e., an integer line search for the discrete choice variables, afterwards multilevel coordinate search (MCS), developed by Huyer & Neumaier [71], for the continuous choice variables and an iteration of this procedure until satisfaction. Thus we hope to find the global optimal solution, but as we are using heuristics there is no guarantee.

4.4 Implementation

In this section we give an implementation of the methods introduced in the last sections. The zero-finding problem mentioned in Section 4.3 is solved numerically by the Matlab package NLEQ, cf. [33], [122].

For a Matlab implementation of the method of separable underestimation we must be able to formulate (4.12). Let `n_o` be n_o, `lZZ(i)` be N_i, let `Ic` correspond to the index set I_c, `Ia` correspond to I_a, let `intcheck` be 1 if $I_d = \emptyset$ (remember we need a constant `c` in this case). Let `objstr`, `viopen`, `constr` be evaluable strings, `objstr` contains the objective function $\sum_{i=1}^{n_o} q_i^T q_i$ as in (4.11), `viopen` contains $\sum_{l \in I_{ia}} (G_{\mathrm{bb}_l} - \sum_{k=1}^{n_o} q_k(\theta_l^k))^2$, and `constr` contains $\sum_{i=1}^{n_o} q_i(\theta_l^i)$ for $l \in I_a$, as in (4.12). Then we can formulate (4.12) by means of CVX [63].

```
cvx_begin
for i=1:n_o
    if any(i==Ic) % i.e., i in I_c
        variable(['q' num2str(i) '(2)']);
    else
        variable(['q' num2str(i) '(' num2str(lZZ(i)) ')']);
    end;
end;
if intcheck
    variable('c(1)');
end;
minimize(eval(objstr)+eval(viopen));
subject to
for i=1:length(Ia)
    eval(constr{Ia(i)})==f(Ia(i));
end;
cvx_end
```

From the solution of this problem the computation of the optimal choice is straightforward by equation (4.14).

As mentioned in the last section an implementation of the strategy using a quadratic model fit is realized by the routine Snobfit available online as Matlab code, and moreover, an implementation of the evolution strategy with covariance matrix adaptation exists as the program CMAES, also available online as Matlab code. Eventually we apply methods of discrete line search and MCS as suggested before, for a limited global search, called `lsearch`.

Let `modelevaluation` be an evaluation of $G_{\text{bb}}(\theta)$ for a given θ. Let `sep` be the implementation of the method of separable underestimation providing a guess `theta` for the optimal design. It may also be replaced or supported by another strategy we introduced, like CMAES or Snobfit. Then the methods described are implemented in a loop as below. A user defined value `optimumconfirmed` defines the stopping criterion: if the best solution does not change for `optimumconfirmed` iterations the loop stops.

```
% load starting points x and corresponding function evaluations f
load startingpoints;

% best starting point
[minf,ind]=min(f);
xbesto=x(ind,:);
fbesto=f(ind);

oc=0;   % needed for the stopping criterion

while 1
    theta=sep(x,f);  % sep replaceable by other strategies

    x=[x;theta];

    % evaluate choice theta
    fnew=modelevaluation(theta);
    f=[f fnew];

    [lsx,lsf]=lsearch(theta);
    x=[x;lsx];
    f=[f lsf];

    % optimal solution
    [minf,ind]=min(f);
    xbest=x(ind,:);
    fbest=f(ind);

    % stopping criterion
    if fbest>=fbesto
        oc=oc+1;
    else
        fbesto=fbest;
        xbesto=xbest;
        oc=1;
    end;
    if oc>=optimumconfirmed
        fbest=fbesto;
        xbest=xbesto;
        break;
    end;
end;
```

The resulting `xbest` is our guess for the optimal design. As already mentioned the heuristics cannot guarantee any global optimality.

Chapter 5

Uncertainty modeling in design optimization

In this chapter we combine uncertainty handling and design optimization formulating our goal of finding the optimal, robust design. The uncertainties will be handled as described in Chapter 3, design optimization will be tackled as in Chapter 4. Similar to the approach in the last chapter we start with the problem formulation in Section 5.1, we point out the characteristics and difficulties that originate from the problem statement in Section 5.2. Using the methods developed we present an approach to a solution for the optimal robust design in Section 5.3.

5.1 Problem formulation

We will stick to the same notation as introduced in the chapters before. As mentioned in Section 4.1 both design and global input variables contained in the vector z are uncertain, ε denotes the related random vector of uncertain errors. We assume that the optimization problem can be formulated as a mixed-integer, **bilevel problem** of the following form:

$$\begin{array}{lll} \min\limits_{\theta} \max\limits_{x,z,\varepsilon} g(x) & & \text{(objective functions)} \\ \text{s.t.} & z = Z(\theta) + \varepsilon & \text{(table constraints)} \\ & G(x,z) = 0 & \text{(functional constraints)} \\ & \theta \in T & \text{(selection constraints)} \\ & V_{\mathcal{T}}(\varepsilon) \leq \underline{V}_{\mathcal{T}_\alpha} & \text{(cloud constraint)} \end{array} \tag{5.1}$$

The table constraints now assign to each choice θ an input vector z whose value is the nominal entry from the table mapping $Z(\theta)$ plus its error ε with uncertainty specified by the cloud. The new **cloud constraint** involves the potential function $V_{\mathcal{T}}$ as described in Chapter 3 and models the worst-case relevant region $\{\varepsilon \in \mathbb{M} \mid V_{\mathcal{T}}(\varepsilon) \leq \underline{V}_{\mathcal{T}_\alpha}\} = \underline{C}_\alpha$ assuming that $\underline{V}_{\mathcal{T}_\alpha}$ exists. The confidence level α should be chosen to reflect the seriousness of consequences of the worst case event. In our applications we used $\alpha = 0.95$, cf. Chapter 7.

With this problem formulation we ask to find the design with the optimal worst-case scenario, that implicates the bilevel structure. It is possible to trade off between the worst-case scenario and the nominal case of a design, but this would lead to a multi-objective optimization problem formulation which will not be investigated in this study.

5.2 Difficulties

Apart from the difficulties already mentioned in Section 5.2 we now face additionally a bilevel structure imposed by the uncertainties, which is already a nontrivial complication in the traditional situation where all variables are continuous. Difficulties arise, e.g., in the evaluation of the objective function, i.e., the inner level. The evaluation can be very expensive as it is an optimization problem itself, and derivatives of the inner level do not exist, so we do not have derivatives in the outer level optimization. The current methods for handling such problems require at least that the objective and the functional constraints are continuously differentiable. Standard optimization tools cannot be used to tackle problem (5.1). Symbolic solvers already had severe problems with solving the one level problem (4.4), so they are not suitable for this even more complicated problem. Currently it is not even possible to formulate problem (5.1) for the real-life applications discussed in Chapter 7 in a symbolic solver language without a modification of the underlying model, cf. Neumaier et al. [118].

Since the design problem (5.1) is so highly complex we are limited to the use of **heuristic** methods and will take advantage of the methods which we already developed in the last chapter.

5.3 Solution approach

The approach to the solution of problem (5.1) now combines all presented methods from uncertainty handling and heuristic design optimization.

We will first reformulate the problem, similarly to (4.7), incorporating the objective function and functional constraints for the underlying model in the black box function

$$G_{\mathrm{bb}}(\theta, \varepsilon) := g(G_o(Z(\theta) + \varepsilon)) = g(x). \tag{5.2}$$

This is possible as we again assume that the functional constraints can be solved numerically for x given z, i.e., $x = G_o(z)$. Then effectively only an optimization over ε is needed in the inner level of the problem.

$$\begin{aligned} \min_{\theta} \quad \max_{\varepsilon} \quad & G_{\mathrm{bb}}(\theta, \varepsilon) \\ \text{s.t.} \qquad & \theta \in T \\ & V_{\mathcal{T}}(\varepsilon) \leq \underline{V}_{\mathcal{T}_\alpha} \end{aligned} \tag{5.3}$$

5.3.1 Remark. The function G_{bb} for a fixed design θ can be interpreted as the design response function s in Chapter 2.

We start with a look at the inner level of the problem, i.e., for a fixed $\theta \in T$

$$\begin{aligned} \max_{\varepsilon} \quad & G_{\mathrm{bb}}(\theta, \varepsilon) \\ \text{s.t.} \quad & V_{\mathcal{T}}(\varepsilon) \leq \underline{V}_{\mathcal{T}_\alpha} \end{aligned} \tag{5.4}$$

Note that the cloud constraint $V_{\mathcal{T}}(\varepsilon) \leq \underline{V}_{\mathcal{T}_\alpha}$ is not linear, so we perform the coordinate transformation $\mathcal{T}$ prior to solving problem (5.4), i.e., $\varepsilon_{\mathcal{T}} = \mathcal{T}(\varepsilon)$, $V_{\mathcal{T}}(\varepsilon) = V_p(\mathcal{T}(\varepsilon)) = V_p(\varepsilon_{\mathcal{T}})$.

Because of the polyhedral structure of V_p, the cloud constraint can then be written as a collection of linear inequalities parameterized by the confidence level α. For a fixed θ we approximate $G_{\mathrm{bb}}(\theta, \mathcal{T}^{-1}(\varepsilon_{\mathcal{T}}))$ linearly in a box b containing the polyhedron $\{\varepsilon_{\mathcal{T}} \in \mathcal{T}(\mathbb{M}) \mid V_p(\varepsilon_{\mathcal{T}}) \leq \underline{V}_{\mathcal{T}_\alpha}\}$ by a function $G_{\mathrm{bb_{lin}}}(\varepsilon_{\mathcal{T}})$. Thus problem (5.4) becomes a linear programming problem (LP),

$$\begin{aligned} \max_{\varepsilon_{\mathcal{T}}} \ & G_{\mathrm{bb_{lin}}}(\varepsilon_{\mathcal{T}}) \\ \text{s.t.} \ & V_p(\varepsilon_{\mathcal{T}}) \leq \underline{V}_{\mathcal{T}_\alpha} \end{aligned} \tag{5.5}$$

The maximizer $\widehat{\varepsilon}_{\mathcal{T}}$ for the fixed design choice θ corresponds to the worst-case objective function value

$$\widehat{G}_{\mathrm{bb}}(\theta) := G_{\mathrm{bb}}(\theta, \mathcal{T}^{-1}(\widehat{\varepsilon}_{\mathcal{T}})). \tag{5.6}$$

Now consider θ not to be fixed. The function $\theta \to \widehat{G}_{\mathrm{bb}}(\theta)$ implicated by the solution of problem (5.5) is the objective function of the outer level of problem (5.3) and can thus be used to get rid of the bilevel structure in (5.3):

$$\begin{aligned} \min_{\theta} \ & \widehat{G}_{\mathrm{bb}}(\theta) \\ \text{s.t.} \ & \theta \in T \end{aligned} \tag{5.7}$$

This 1-level problem can be solved with the same techniques like problem (4.7), cf. Section 4.3 and Section 4.4, seeking the robust, optimal design.

5.3.2 Remark. By approximating G_{bb} linearly we assume that G_{bb} is a nice behaving function for fixed θ. Nice behaving means that there are no strong peaks within the box b, that G_{bb} has little curvature, or is monotone in b. Thus a linearization would be justified. This is the case in many real-life situations if the box b is reasonably small.

For the implementation in Section 4.4 the `modelevaluation` has to be replaced by a worst-case search within the polyhedron $\{\varepsilon_{\mathcal{T}} \in \mathcal{T}(\mathbb{M}) \mid V_p(\varepsilon_{\mathcal{T}}) \leq \underline{V}_{\mathcal{T}_\alpha}\}$ to represent $\widehat{G}_{\mathrm{bb}}$. But this worst-case search is a straightforward solution of the LP (5.5), using CVX.

Chapter 6

CBDO software

In this chapter we present the realization of our methods in a **graphical user interface** (GUI). Since our approach can be considered as design optimization based on uncertainty modeling with clouds, we call the software **cloud based design optimization** (CBDO), cf. [19].

We have developed the GUI using the concept as already visualized in Figure 2.6, i.e., we have a sequential, iterative structure. The first two steps represent the uncertainty elicitation, that is on the one hand, providing the initially available information (Section 6.1), on the other hand, scenario exclusion (Section 6.2). The next step is the optimization phase (Section 6.3). Eventually, the user has the possibility of adaptively refining the uncertainty information and rerunning the procedure until satisfaction (Section 6.4).

To install the software it is sufficient to copy the `cbdo` folder on the computer where it is going to be run, change to that folder in Matlab, run `cvxsetup` once to set up CVX. Afterwards one calls the program by `cbdogui`, it should work out of the box. To prepare the Matlab file containing the underlying system model $G(x, z)$ (cf. Chapter 4) one has to use the following conventions. Choose a name for the model file, e.g., `model.m`, and create the model file to have the form

```
[obj,rekvdiff]=model(rekvars,uncvars,designvars,fixedvars)
```

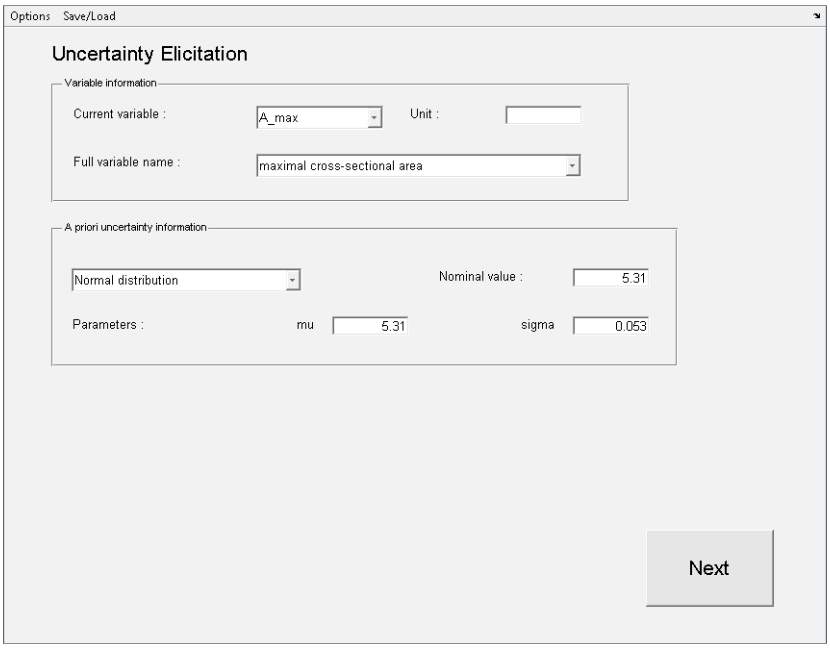
where rekvars corresponds to x, uncvars to u, designvars to v, fixedvars to a set of constant parameters, obj to $g(x)$, and rekvdiff to $G(x,z)$. The inputs of the model file have to be declared in a Matlab script file, say, varinit.m. Both model.m and varinit.m have to be created in the /model subfolder. This subfolder also contains templates for the inputs declaration file, i.e., /model/samplevarinit.m, and for the underlying model file, i.e., /model/samplemodel.m.

6.1 Uncertainty elicitation

After starting the GUI with cbdogui it asks whether to load the last state to the workspace unless it is run for the first time. In the latter case one should first configure the options to set up the model file and inputs declaration file names, and other user-defined parameters after clicking *Options/Edit Options*. The notation from the Chapters 3, 4, and 5 in correspondence to the editable options, and their associated default values are shown in Table 6.1. Tooltips are given for each option in the GUI to guide the user through the configuration. Having set up the options one returns to the uncertainty elicitation clicking *Back*. The initially available information specified in the inputs declaration file can be modified choosing a variable's name and specifying its associated marginal CDF, or interval bound, respectively, cf. Figure 6.1.

Table 6.1: Correspondence of notation

GUI notation	**Symbolic notation**	**Default value**
Samplesize	N_S	1000
Transformation	$\mathcal{T}$	Logarithmic
Confidence level (cloud)	α, cf. (5.1)	95%
Confidence level (sample)	α_S	99.8%
Confidence level (weights)	α_w	95%
Confidence level (width)	α_b	95%
Weight computation constraints	N_{CVX}	3000
Number of starting points	N_o	$\binom{n_o+3}{2}$
Optimum confirmations	optimumconfirmed	2



Uncertainty Elicitation

Variable information

Current variable : A_max

Unit :

Full variable name : maximal cross-sectional area

A priori uncertainty information

Normal distribution

Nominal value : 5.31

Parameters : mu 5.31 sigma 0.053

Figure 6.1: Example for the uncertainty elicitation GUI.

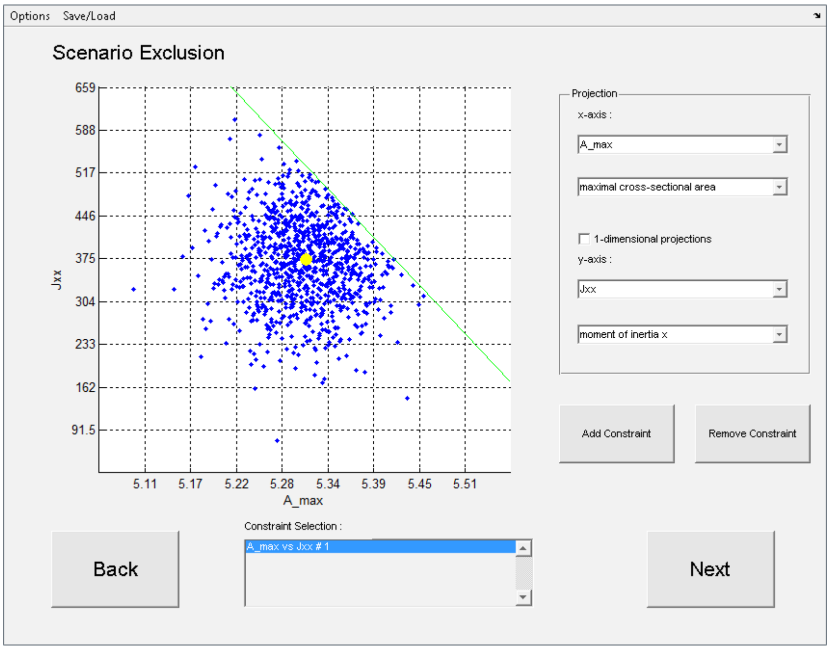
The *Next* button leads to the next step which is scenario exclusion.

6.2 Scenario exclusion

From the information given in the first step the program generates a sample as described in Section 3.3. The second step enables the user to exclude scenarios by linear constraints as described in Section 3.4 and Section 5.3.

To this end the user selects a 1-dimensional or 2-dimensional projection of the generated sample using the field *Projection* on the right. To add a constraint one hits the *Add constraint* button and defines the linear exclusion by two clicks into the sample projection on the left. All linear constraints can be selected from the *Constraint Selection* box to revise and possibly remove them via the *Remove constraint* button. Figure 6.2 shows a possible exclusion in two dimensions.

After the exclusion the *Next* button leads to the optimization phase.



x-axis : A_max

maximal cross-sectional area

□ 1-dimensional projections

y-axis : Jxx

moment of inertia x

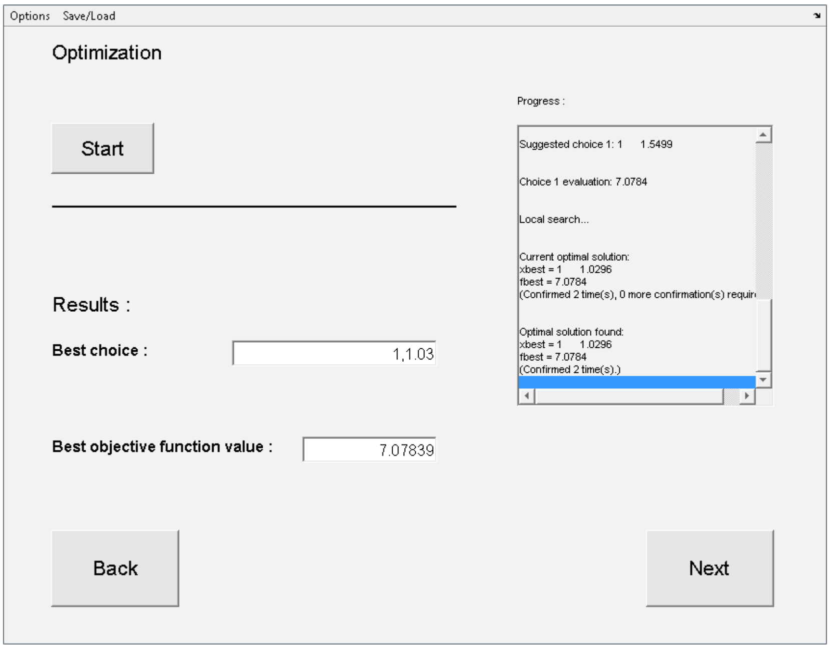
Constraint Selection : A_max vs Jxx # 1

Figure 6.2: Example for scenario exclusion.

6.3 Optimization

The *Start* button initiates two computations: potential cloud generation, which includes weight computation if necessary, cf. Section 3.3, and optimization, cf. Chapter 5. As a result one gets the optimal design point found by the program, and the associated objective function value, cf. Figure 6.3. It should be remarked that the workspace of the optimization including all results is stored as .mat files in the `cbdo` directory.

The user now has the possibility for the adaptive analysis of the results as indicated in Chapter 3. Thus the *Next* button leads back to the uncertainty elicitation which is waiting for refinement.



Best choice : 1,1.03

Best objective function value : 7.07839

Progress :

```
Suggested choice 1: 1      1.5499

Choice 1 evaluation: 7.0784

Local search...

Current optimal solution:
xbest = 1      1.0296
fbest = 7.0784
(Confirmed 2 time(s), 0 more confirmation(s) requir

Optimal solution found:
xbest = 1      1.0296
fbest = 7.0784
(Confirmed 2 time(s).)
```

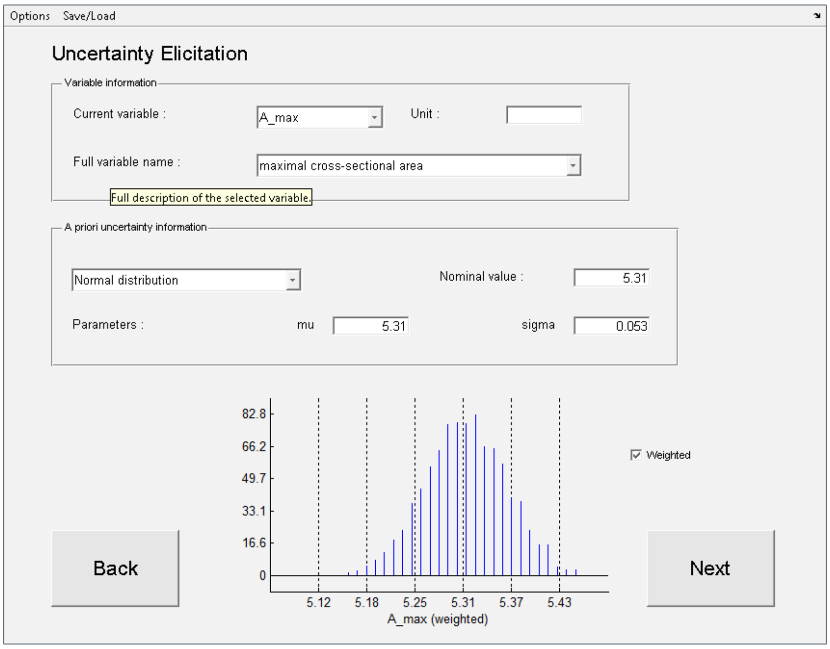
Figure 6.3: Example for the optimization phase.

6.4 Adaptive step

The GUI determining the *a priori* information is not modifiable anymore at this stage of the program. Meanwhile, observe that in the lower part of the GUI a histogram illustrates the weighted marginal distributions of the sample, which respect both the *a priori* information and the scenario exclusion, cf. Figure 6.4.

Hitting the *Next* button makes the scenario exclusion appear again and enables the *a posteriori* adaption of the uncertainty information. For example, the expert who uses the program can consider the worst-case scenario (highlighted with a red dot) to be too pessimistic and exclude it, cf. Figure 6.5.

The *Next* button leads to the optimization phase again and the user can rerun the procedure until satisfaction.



Uncertainty Elicitation

Variable information

Current variable : A_max

Unit :

Full variable name : maximal cross-sectional area

Full description of the selected variable.

A priori uncertainty information

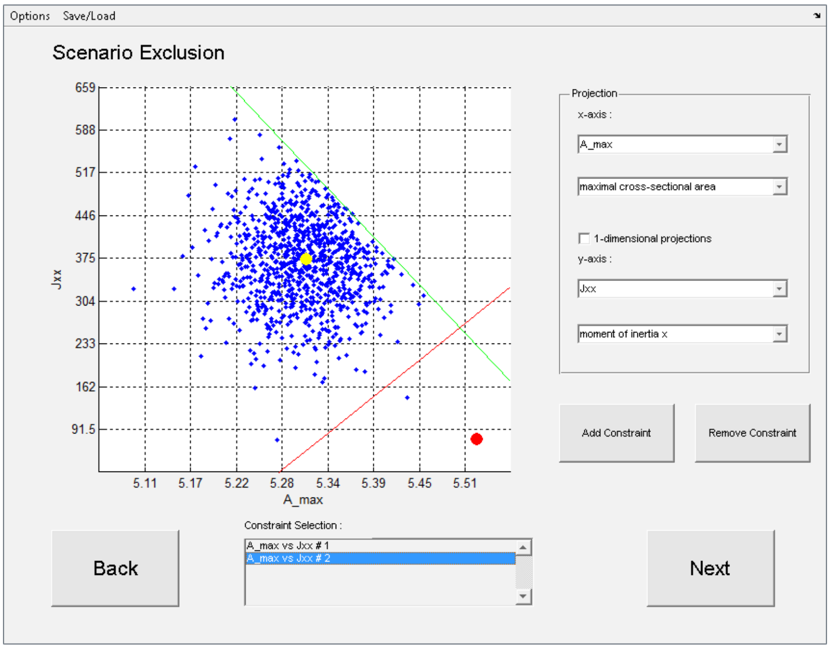
Normal distribution

Nominal value : 5.31

Parameters : mu 5.31 sigma 0.053

☑ Weighted

Figure 6.4: Example for uncertainty elicitation in the adaptive step.



Scenario Exclusion

Projection

x-axis : A_max

maximal cross-sectional area

□ 1-dimensional projections

y-axis : Jxx

moment of inertia x

Constraint Selection :

A_max vs Jxx # 1

A_max vs Jxx # 2

Figure 6.5: Example for *a posteriori* scenario exclusion.

Chapter 7

Space mission design applications

We already mentioned a few applications of uncertainty handling in Chapter 1 and Chapter 2. In this chapter we refer to further studies that can be found in the literature, applying uncertainty models in real-life situations. The main focus of this chapter are two examples of successful application of our approach to robust design optimization based on clouds, cf. Section 7.1 and Section 7.2.

Safety analysis using advanced methods for uncertainty handling has numerous, diverse fields of application, cf. Sexsmith [143]: risk mitigation for new and existing bridge structures, avoiding of ship collisions, nuclear power plant siting. But also less engineering related topics involve uncertainty handling nowadays. A comparison of the success of different teaching techniques in education with uncertainty methods can be found in Kosheleva & Ceberio [83]. Survival analysis and prediction of epidemics is researched in Bickis & Bickis [15]. Problems from food processing concerned with the ripening of cheese are studied in Baudrit et al. [5]. The problem of jury compositions in jurisprudence is investigated with statistical methods in Coolen et al. [23].

Space related sciences have employed uncertainty methods as well, such as trajectory optimization in formation flying spacecrafts [136]. The two highlighted examples presented in detail in the next sections are typical problems found in the field of spacecraft system design. In the past of spacecraft engineering there have been only small steps made in quantifying reliability in high-dimensional design problems. The conventional approach is based on simulation techniques [4], [97], [150], or on cautious attempts to involve more sophisticated methods, e.g., from fuzzy logic, cf. Ross [138], or evidence theory, cf. Croisard et al. [26], with the drawbacks introduced in Chapter 2. To harmonize the different disciplines involved in spacecraft design one follows the approach of multidisciplinary design optimization traditionally using safety margins as uncertainty tool, cf. Section 2.1.4.

In the following sections we investigate two case studies with respect to the applicability of our cloud based design optimization approach to real-life problems with high-dimensional uncertainties given incomplete information. In Section 7.1 we study the XEUS mission which leads to a 10-dimensional design problem with a 24-dimensional uncertainty domain. In Section 7.2 we study the Mars Exploration Rover mission stating a 1-dimensional design problem with a 34-dimensional uncertainty domain. We produce and assess the results and compare them to the conventional safety margins methods popular among many engineers. We did not choose a transformation $\mathcal{T}$ in these examples, and the experts did not provide dependency constraints. A publication of the results can also be found in Fuchs et al. [58] and Neumaier et al. [118].

7.1 Example: XEUS mission

In this section we investigate a spacecraft design problem in the context of the 2004 X-ray Evolving Universe Spectroscopy (XEUS) mission of the European Space Agency (ESA). The XEUS is supposed to be a permanent space-borne X-ray observatory with the scientific goal to study black holes, galaxy groups, clusters, and the interstellar medium, cf. [43], [168].

Provided by ESA with Matlab versions of modified subsystem models for the XEUS mission (not the original models), our goal is to find a spacecraft design for the given subsystems considering minimization of the total mass m_{tot} and robustness of the solution. The modified version of the models, implemented on the basis of Larson & Wertz [93], consists of nine m-files defining Matlab functions: six for the various subsystems, one for the mass budget, and two auxiliary functions, cf. Table 7.1. The uncertainty information was given as interval uncertainty on the vector of global input variables u, cf. Chapter 4, in 24 dimensions, which is a representative dimensionality for real-life uncertainty handling. The vector of design choices θ is 10-dimensional, where $\theta^1, \dots, \theta^9$ are discrete choices (cf. Appendix A.1), and $\theta^{10} \in [0.5, 8.0]$ is a continuous choice. This is a very complex setting for a design problem. The functional constraint $G(x, z) = 0$ is given as a fixed point problem, i.e., $\overline{G}(x, z) = x$, hence $G(x, z) = \overline{G}(x, z) - x$, where x is the vector containing the three recursive variables in the model: m_{tot}, `pow_prop` and `pow_ther`.

Table 7.1: Subsystem models

file name	**use**
`DH_ss.m`	Data Handling System
`TTC_ss.m`	Tracking, Telemetry and Command
`pow_ss.m`	Power System
`str_ss.m`	Structure System
`str_stabil.m`	aux. for Structure System
`target_planet_func.m`	aux. for Thermal Protection System
`thermal_ss.m`	Thermal Protection System
`prop_ss.m`	Propulsion System
`mass_budget.m`	Mass Budget

The variable structure is summarized in Appendix A.1. An evaluation of $\overline{G}(x, z)$ amounts to running once the Matlab functions for the subsystems, cf. Appendix A.2.

7.1.1 Results

The cloud constraint for the optimization is generated for a confidence level of $\alpha = 95\%$ and a generated sample size $N_S = 1000$. For the optimization we used heuristics based on multiple runs with Snobfit. The best design point found is given in Table 7.2, showing the optimal design choice and the corresponding value of the objective function m_{tot} for the nominal case and for the worst case, respectively. In total we made 20 runs considering the uncertainty in the

Table 7.2: Nominal and worst-case values of m_{tot} for the optimal design choice

Design Choice θ	$(27, 4, 9, 33, 14, 1, 1, 2, 1, 0.6125)$
Nominal value m_{tot}	1566.1
Worst-case m_{tot}	1730.7

model inputs, and 20 additional runs not taking uncertainties into account. Figure 7.1 illustrates the difference of the results computed with and without uncertainty. By comparing with the diagonal line $x = y$, we see that for the robust solutions found (crosses running almost parallel), the difference between the worst-case and the nominal case for m_{tot} appears to be almost constant, reflecting the effect of the variation in the domain of uncertainty.

On the other hand, only two of the choices found without uncertainty methods (circles) are feasible under all admissible uncertainties; and in these two cases, the worst-case value in the feasible range is much worse than that for the robust solutions found.

Moreover, by comparing the values for the nominal case (the x values) for both crosses and circles, we see that the best robust solutions (found by optimization under uncertainty) have nominal values of m_{tot} which are competitive with those of the best solutions computed without uncertainty.

We conclude that in this application example the quality of the optimal solutions does not differ significantly, whether or not you take uncertainty into account, while the robustness is drastically improved.

We also investigate the behavior of the objective function within the given uncertainty domain: for a robust choice in Figure 7.3, for an unstable choice in Figure 7.2, disclosing various hidden constraints. In general, hidden constraints such as those in Figure 7.2 (gaps in $\mathbb{R}^{24}$) cannot be handled with heuristics.

In conclusion we are able to solve the robust optimization problem in case of interval uncertainties satisfactorily except for the presence of hidden constraints, which could possibly be avoided by suitable attention during the creation of the subsystem models.

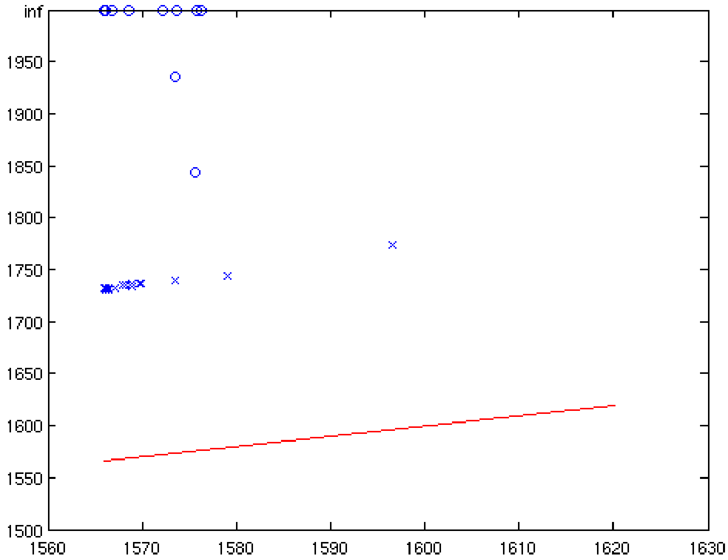


Figure 7.1: Circles denote design points found in different runs by optimization without uncertainty; crosses denote design points found by optimization with uncertainty. The x-axis displays the nominal value of m_{tot} for each design point, while the y-axis displays the worst-case value of m_{tot} for each design point. Thus we plot the points $(x(\theta_i),y(\theta_i))$ for each design point θ_i. Because of the different ranges of the coordinates, the points where $x = y$ lie on the only slightly slanted line drawn. The line should help to see the difference between the worst-case and the nominal value for m_{tot}.

Solving the optimization problem with uncertainty revealed significant robustness advantages of our approach using uncertainty, without significantly compromising the quality of the solutions at the nominal values.

7.2 Example: Mars Exploration Rover

In this section we apply our methods to the Attitude Determination and Control Subsystem (ADCS) for the NASA's Mars Exploration Rover (MER) mission [42], [99] whose scientific goal is to investigate the history of water on

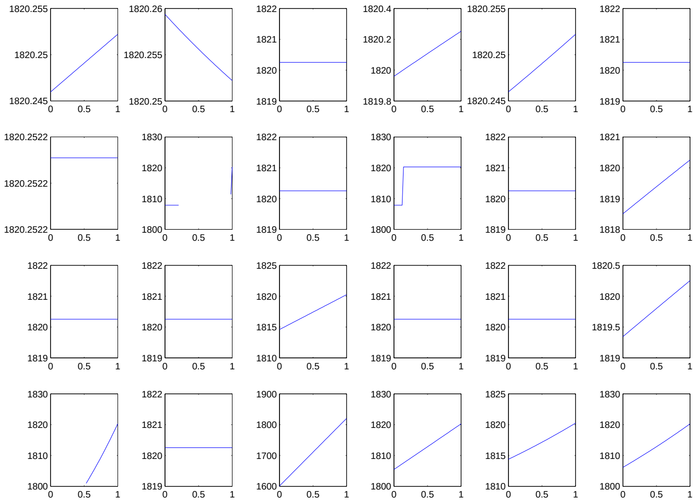


Figure 7.2: For each uncertain variable u^i, $i = 1, \ldots, 24$ (scaled) we fix all the other uncertain variables $u^k, k \neq i$ at the midpoint of their boxes and plot $m_{tot}(u^i)$. The figure discloses various hidden constraints.

Mars. The ADCS is composed by eight thrusters aligned in two clusters. Onboard the spacecraft there is no main propulsion subsystem. The mission sequence after orbit injection includes a number of spin maneuvers and slew maneuvers. Spin maneuvers are required for keeping the gyroscopic stability of the spacecraft, whereas slew maneuvers serve to control the direction of the spacecraft and to fight effects of solar torque. Fault protection is considered to correct possible errors made when performing nominal maneuvers.

Our goal is to select the type of thrusters (from a set of possible candidates as listed in Table B.3) considering both minimization of the total mass m_{tot}, and assessment of the worst possible performance of a thruster with respect to m_{tot}. That corresponds to finding the thruster with the minimal worst-case scenario, i.e., a problem formulation as in Chapter 5. The total mass consists of the fuel needed for attitude control (computed as the sum of the fuel needed for each maneuver) plus the mass of the eight thrusters that need to be mounted on the spacecraft. According to our notations, the choice variable θ, i.e., the type of thruster, can be selected as an integer between 1 and 30.

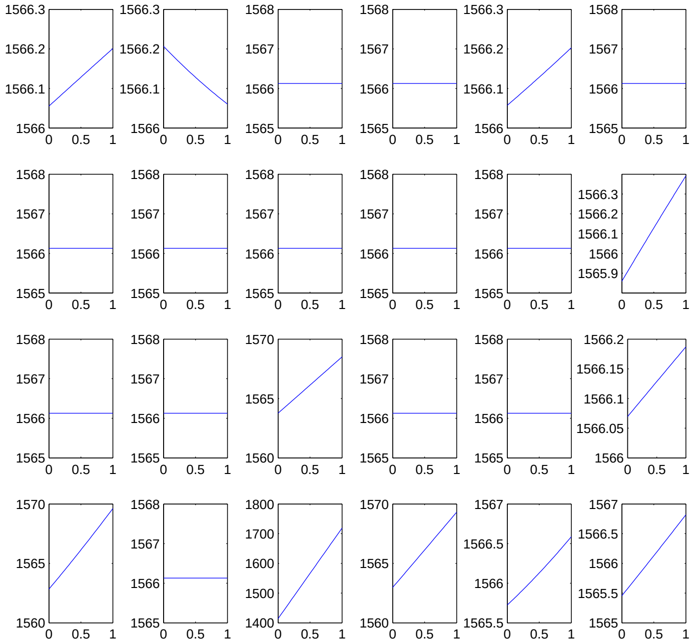


Figure 7.3: The same information as in Figure 7.2, but for a robust choice. Within the domain of uncertainty, the dependence on the uncertain variables is essentially linear.

Uncertainty specifications, variable structure, the MER mission maneuver sequence, and system model equations to compute the total mass m_{tot} are taken from Thunnissen [156]. The uncertainty specification for the model variables are reported in Table B.4 in Appendix B.5. The number of uncertain global input variables (dimension of u) in this application example is 33 plus 1 uncertain design variable. The variable structure is summarized in Appendix B.1. Moreover, a survey on the system model equations and the MER mission sequence can be found in Appendix B.2 and Appendix B.3, respectively.

Compared to the case study in Section 7.1 this example is even more complicated from the uncertainty handling point of view, i.e., we have to cope with a higher dimensionality, and additional uncertainty information in form of given marginal CDFs.

7.2.1 Results

The cloud constraints for the optimization are generated for a confidence level of $\alpha = 95\%$ and a generated sample size $N_S = 1000$. We produce results for four different settings of uncertainty information and handling:

a. The uncertainties are as specified in Table B.4. Here we treat them in a classical engineering way, assigning $3\,\sigma$ (i.e., 3·standard deviation) safety margins to the uncertain variables which is supposed to correspond to a 99.7% confidence interval for a single variable. Then the optimal design choice is $\theta = 9$ with an objective function value of $m_{tot} = 3.24$ kg in the nominal case, and $m_{tot} = 5.56$ kg in the worst case.

b. The uncertainties are again as in Table B.4. With our methods we find the optimal design choice $\theta = 9$ as in setting (a). However, if we compare the worst-case analysis of b and a, it is apparent that the results for the 3 σ boxes are far too optimistic to represent a reliable worst-case scenario, the value of m_{tot} is now 8.08 kg instead of 5.56 kg for the 3 σ boxes.

c. In this setting we do not take any uncertainties into account, generally assuming the nominal case for all uncertain input variables for the optimization. The optimal design choice then is $\theta = 3$ with a value of $m_{tot} = 2.68$ kg in the nominal, but $m_{tot} = 8.75$ kg in the worst case, which is significantly worse than in setting (b).

d. The uncertainties are obtained by taking the values from Table B.4 and doubling the standard deviation of the normally distributed variables. It is interesting to report that if we increase the uncertainty in the normally distributed uncertain variables simply in this way, the optimal design choice changes to $\theta = 17$ with a value of $m_{tot} = 3.38$ kg in the nominal, and $m_{tot} = 9.49$ kg in the worst case.

The results are summarized in Table 7.3, showing the optimal design choice for each setting and the corresponding value of the objective function m_{tot} for the nominal case and for the worst case, respectively. The results show a number of important facts related to spacecraft design. The comparison between the settings (b) and (d) suggests that the optimal design point θ is quite sensitive to the uncertainty description, a fact well-known to the system engineers who see their spacecraft design changing frequently during preliminary phases when

Table 7.3: Nominal and worst-case values of m_{tot} for different design choices obtained by the four different settings.

Setting	**Design Choice** θ	**Nominal value** m_{tot}	**Worst-case** m_{tot}
a	9	3.24	5.56
b	9	3.24	8.08
c	3	2.68	8.75
d	17	3.38	9.49

new information becomes continuously available. Our method captures this important dynamics and processes it in rigorous mathematical terms.

The comparison between the settings (b) and (c) suggests that the uncertainties need to be accounted for in order not to critically overestimate the spacecraft performances.

Finally, the comparison between the settings (b) and (a) suggests that the simple 3 σ analysis of uncertainties, frequent in real engineering practice, produces quite a different estimation of the spacecraft performances with respect to a more rigorous accounting of the uncertainty information.

Appendix A

XEUS case study

The following sections report the variable structure, the underlying subsystem model function calls for the application example from Section 7.1.

Notation conflicts between some physical variables in the model and the variables we used in previous chapters are possible, however, their meaning should be clear from the context.

A.1 Variable structure

The 107 variables involved in the subsystem functions fall into five different classes:

- **24 fixed variables**.
  Input variables for the functions with fixed values and no uncertainty. Among them are a number of variables which have the value zero, which arise since the problem under discussion is in fact part of a bigger model in which these variables would get nontrivial values.

- **24 global input variables**.
  Uncertainty variables which correspond to the vector u in Chapter 5.

- **24 design variables**.
  The 24 design variables belong to 10 different design choices $\theta^1, \dots, \theta^{10}$,

each of which can be chosen independently and determines one or more design variables. The choices $\theta^1, \dots, \theta^9$ are discrete (cf. Table A.1), and $\theta^{10} \in [0.5, 8.0]$ is a continuous choice.

Table A.1: Choices θ^i and size of the corresponding table τ^i, i.e., $N_i \times n_i$

choice variable	**size of τ^i**
θ_1	4×3
θ_2	14×1
θ_3	6×1
θ_4	8×1
θ_5	5×4
θ_6	20×3
θ_7	9×4
θ_8	44×2
θ_9	30×4

- **16 only result variables**.
  The only result variables only occur as result variables and do not contain the objective function. Hence they are immaterial for the difficulty of the problem.

- **19 intermediate result variables**.
  All other variables are intermediate result variables, and occur both as input of some file and as output of some file.

For the detailed description of the values and interval bounds for the fixed variables and global inputs, respectively, and for the full tables τ^i, see Neumaier et al. [118].

A.2 Subsystem function calls

The Matlab functions for the subsystems are called as follows:

```
[mass_DH,pow_DH,cost_DH,aff_DH]=...
        DH_ss_preprocessed(data_tot,punt_mem,punt_mem_mass,...
        punt_mem_pow)

[beam,mass_TTC,pow_TTC,cost_TTC,aff_TTC]=...
        TTC_ss_preprocessed(punt_f,punt_D,punt_Eb,elev,S,GSD,DR)

[diam_SA,P_SA,A_SA,T_SA,mass_SA,cost_SA,aff_SA,capacita,...
volume_batt,mass_batt,cost_batt,aff_batt,pow_tot,...
mass_pow,cost_pow,aff_pow]=...
        pow_ss_preprocessed(punt_eta,punt_alfa,punt_ro,punt_d,...
        punt_spec,punt_eff,punt_dens,max_sun_dist,min_sun_dist,...
        Tecl,teta0,Tday,y,cicli,body_mount_SA,primary,...
        target_planet,punt_h_vs_r,powzero,pow_TTC,pow_DH,...
        pow_aocs,pow_prop,pow_ther)

[diam,S_Thickness,height,area_tot,mass_str,cost_str,aff_str]=...
        str_ss_preprocessed(punt_El,punt_rho,punt_ult_str,...
        punt_h_vs_r,punt_yie_str,m_tot,diam_SA,body_mount_SA,...
        ax_g,lat_g,ax_freq,lat_freq)

[H_planet,Gs_hot_2,H_min_IR]=...
        target_planet_func2_preprocessed(target_planet,...
        H_terra,H_target)

[rad_area,mass_ther,pow_ther,c_ther,a_ther]=...
        thermal_ss_preprocessed(Tup,Tdown,target_planet,...
        min_sun_dist,max_sun_dist,punt_alfa_sup,punt_eps_sup,...
        area_tot,A_SA,P_SA,dens_dep_rad,Tecl,body_mount_SA,...
        m_tot,H_planet,Gs_hot_2,H_min_IR)

[m_fuel,m_tot_prop,mass_prop,pow_prop,N_prop,cost_prop,...
aff_prop,cost_tot,aff_tot]=...
        prop_ss_preprocessed(deltaV,T_PP,punt_I,punt_m_eng,...
        punt_P_eng,punt_T_eng,mzero,mass_TTC,mass_DH,...
        mass_aocs,mass_pow,mass_ther,mass_str)

[mass_harness,m_tot,m_tot_mb]=...
        mass_budget_preprocessed(mzero,mass_TTC,mass_pow,...
        mass_prop,mass_str,mass_DH,mass_aocs,mass_ther,m_fuel,...
        count)
```

Appendix B

MER case study

The following sections report the variable structure, the underlying model equations, the mission maneuver sequence, the design and the uncertainty specifications for the application example from Section 7.2.

Notation conflicts between some physical variables in the model and the variables we used in previous chapters are possible, however, their meaning should be clear from the context.

B.1 Variable Structure

The 49 variables involved in the model fall into the following four categories:

- **7 fixed parameters**.
  Input variables for the model with fixed values and no uncertainty (for the values see Table B.1).
  1. c_0, speed of light in a vacuum
  2. d, average distance from the spacecraft to the sun in AU
  3. g_0, gravity constant
  4. t, total mission time
  5. θ_i, sunlight angle of incidence

6. χ, the specific impulse efficiency parameter is a property of a particular thruster. Lacking the specification of χ for several thrusters we fixed χ to take the same values for all thrusters.
7. c_1, the numerical solution x of $\tan(x) - x/(1-\chi) = 0$

Table B.1: Values of the fixed parameters

Fixed parameter	**Value**
c_0	$3 \cdot 10^8$ m/s
d	1.26 AU
g_0	9.81 m/s^2
t	216 days
θ_i	0°
χ	0.0375
c_1	0.334

- **33 Uncertain input variables**.
  The uncertainties are specified by probability distributions for each of these variables (cf. Appendix B.5).
  1. A_{max}, maximal cross-sectional area
  2. J_{xx}, J_{zz}, moments of inertia
  3. R, engine moment arm
  4. δ_1, δ_2, engine misalignment angle
  5. g_s, solar constant at 1 AU
  6. κ, distance from the center of pressure to the center of mass
  7. ω_{spin_i}, spin rates, $i = 0...3$, given in rpm
  8. ψ_{slew_i}, slew angles, $i = 1...19$, given in °
  9. q, spacecraft surface reflectivity
  10. $uncfuel$, additive uncertain constant that represents inaccuracies in the equations used for the calculation of the fuel masses

- **3 Design variables**.
  Thruster specifications relevant for the model. There is uncertainty information given on one of them (the thrust).
  1. F, thrust
  2. I_{sp}, specific impulse

3. m_{thrust}, mass of a thruster

- **6 Output variables**.
  Result variables containing the objective for the optimization m_{tot}.
  1. m_{fp}, fuel mass needed for fault protection maneuvers
  2. m_{fuel}, total fuel mass needed for all maneuvers
  3. m_{slew}, fuel mass needed for slew maneuvers
  4. m_{slew_s}, fuel mass needed for slew maneuvers fighting solar torque
  5. m_{spin}, fuel mass needed for spin maneuvers
  6. m_{tot}, total mass of the subsystem

B.2 Model equations

The background for the equations of the ADCS subsystem model are the equations from Chapter 9 of Thunnissen [156].The basic equations are as follows:

$$c = I_{sp} \cdot g_0 \tag{B.1}$$
$$r = \sin(40°) \cdot R \tag{B.2}$$
$$F_{ideal_{tot}} = 2 \cdot F \tag{B.3}$$
$$F_{act_{tot}} = (\cos(\delta_1) + \cos(\delta_2)) \cdot F \tag{B.4}$$

To calculate the fuel mass m_{spin} needed for one spin maneuver (change in spin rate from ω_{spin_i} to $\omega_{spin_{i+1}}$, $i = 0...2$) the following equations are given:

$$\Delta\omega_{ideal} = |\omega_{spin_i} - \omega_{spin_{i+1}}| \tag{B.5}$$
$$I_{ideal} = \frac{\Delta\omega_{ideal} \cdot J_{zz}}{r} \tag{B.6}$$
$$t_{spin} = \frac{I_{ideal}}{F_{ideal_{tot}}} \tag{B.7}$$
$$I_{actual} = t_{spin} \cdot F_{act_{tot}} \tag{B.8}$$
$$m_{spin} = \frac{I_{actual}}{c} \tag{B.9}$$

To calculate the fuel mass m_{slew} needed for one slew maneuver the following equations are given (requires the slew angle ψ_{slew} for the maneuver and the current spin rate ω_{spin} at the time the maneuver is performed):

$$t_{half\_rev} = \frac{\pi}{\omega_{spin}} \tag{B.10}$$

$$t_{on_{ideal}} = \frac{2 \cdot c_1}{\omega_{spin}} \tag{B.11}$$

$$\Delta\phi_{ideal} = t_{on_{ideal}} \cdot \omega_{spin} \tag{B.12}$$

$$\Delta\tau = \frac{2 \cdot F_{ideal_{tot}} \cdot r}{\Delta\phi_{ideal} \cdot \sin\left(\frac{\Delta\phi_{ideal}}{2}\right)} \tag{B.13}$$

$$H = J_{zz} \cdot \omega_{spin} \tag{B.14}$$

$$\Delta\psi_{ideal} = \frac{\Delta\tau \cdot t_{on_{ideal}}}{H} \tag{B.15}$$

$$n_{pulses_{ideal}} = \left\lceil \frac{\psi_{slew}}{\Delta\psi_{ideal}} \right\rceil \tag{B.16}$$

$$\Delta\psi = \frac{\psi_{slew}}{n_{pulses_{ideal}}} \tag{B.17}$$

$$\Delta I_{torque} = H \cdot \Delta\psi \tag{B.18}$$

$$\Delta\phi = 2 \cdot \arcsin\left(\frac{\Delta I_{torque} \cdot \omega_{spin}}{2 \cdot F_{ideal_{tot}} \cdot r}\right) \tag{B.19}$$

$$t_{on} = \frac{\Delta\phi}{\omega_{spin}} \tag{B.20}$$

$$\eta = \frac{t_{on}}{t_{half\_rev}} \tag{B.21}$$

$$c_{sd} = c \cdot \eta^{\chi} \tag{B.22}$$

$$m_{slew} = n_{pulses_{ideal}} \cdot \frac{F_{act_{tot}} \cdot t_{on}}{c_{sd}} \tag{B.23}$$

To calculate the total fuel mass m_{fuel} needed for all maneuvers we compute for each maneuver to be performed the mass m_{spin} or m_{slew} (depends on the maneuver type), and achieve m_{fuel} as the sum of these masses. To calculate the total mass m_{tot} we compute

$$m_{tot} = m_{fuel} \cdot (1 + uncfuel) + 8 \cdot m_{thrust} \tag{B.24}$$

B.3 MER mission sequence

The sequence of maneuvers for the MER mission is listed in Table B.2.

B.4 Thruster specification

Table B.3 shows the thruster specifications and the linked choice variable θ. The table entries are sorted by the thrust F. The difference between the so-called design and choice variables can be seen easily in this table (τ^1): the table is a $N_1 \times n_1$ table with $N_1 = 30$, $n_1 = 3$, and thus represents 30 discrete choices in $\mathbb{R}^3$. The 3 design variables v_1^1, v_2^1, v_3^1 are the 3 components of these points in $\mathbb{R}^3$. The choice variable $\theta = \theta^1$ is 1-dimensional ($n_o = 1$) and has an

Table B.2: Mission sequence (cf. Thunnissen [156])

Mission sequence event	Maneuver type	parameter	value	unit
De-spin from 3^{rd} stg.	spin	ω_{spin1}	2.000	rpm
A-practice	slew	ψ_{slew1}	5.000	°
ACS-B1	slew	ψ_{slew2}	50.45	°
ACS-B2	slew	ψ_{slew3}	5.130	°
ACS-B3	slew	ψ_{slew4}	6.350	°
ACS-B4	slew	ψ_{slew5}	2.760	°
ACS-B5	slew	ψ_{slew6}	8.510	°
ACS-B6	slew	ψ_{slew7}	9.880	°
ACS-B7	slew	ψ_{slew8}	5.640	°
ACS-B8	slew	ψ_{slew9}	5.040	°
ACS-B9	slew	ψ_{slew10}	5.750	°
ACS-B10	slew	ψ_{slew11}	4.470	°
ACS-B11	slew	ψ_{slew12}	5.530	°
ACS-B12	slew	ψ_{slew13}	5.850	°
FP: spin event	spin	ω_{spin2}	2.750	rpm
FP: spin recovery	spin	ω_{spin3}	7.410	rpm
FP: emergency slew 1	slew	ψ_{slew14}	15.75	°
FP: emergency slew 2	slew	ψ_{slew15}	15.75	°
FP: emergency slew 3	slew	ψ_{slew16}	15.75	°
FP: emergency slew 4	slew	ψ_{slew17}	15.75	°
FP: emergency slew 5	slew	ψ_{slew18}	15.75	°
FP: emergency slew 6	slew	ψ_{slew19}	15.75	°

integer value between 1 and 30. The various sources for the data contained in Table B.3 are [36], [128], [132], [156], [174].

B.5 Uncertainty specification

All uncertainty specifications taken from Thunnissen [156] are reported in Table B.4. The notation used for the distributions is as in Table 2.1. The uncertainty information on the design variable F should be interpreted as follows: The actual thrust of a thruster is normally distributed, has the mean F_{table} (:= the nominal value for F specified in Table B.3) and standard deviation $\frac{7}{300}F_{table}$.

Table B.3: Thruster specifications and the linked choice variable θ, given as a table τ^1 with $N_1 = 30$, $n_1 = 3$: Thrust F in Newtons, specific impulse I_{sp} in seconds, mass m_{thrust} in grams

θ	**Thruster**	F	I_{sp}	m_{thrust}
1	Aerojet MR-111C	0.27	210.0	200
2	EADS CHT 0.5	0.50	227.3	200
3	MBB Erno CHT 0.5	0.75	227.0	190
4	TRW MRE 0.1	0.80	216.0	500
5	Kaiser-Marquardt KMHS Model 10	1.0	226.0	330
6	EADS CHT 1	1.1	223.0	290
7	MBB Erno CHT 2.0	2.0	227.0	200
8	EADS CHT 2	2.0	227.0	200
9	EADS S4	4.0	284.9	290
10	Kaiser-Marquardt KMHS Model 17	4.5	230.0	380
11	MBB Erno CHT 5.0	6.0	228.0	220
12	EADS CHT 5	6.0	228.0	220
13	Kaiser-Marquardt R-53	10	295.0	410
14	MBB Erno CHT 10.0	10	230.0	240
15	EADS CHT 10	10	230.0	240
16	EADS S10 - 01	10	286.0	350
17	EADS S10 - 02	10	291.5	310
18	Aerojet MR-106E	12	220.9	476
19	SnM 15N	15	234.0	335
20	TRW MRE 4	18	217.0	500
21	Kaiser-Marquardt R-6D	22	295.0	450
22	Kaiser-Marquardt KMHS Model 16	22	235.0	520
23	EADS S22 - 02	22	290.0	650
24	ARC MONARC-22	22	235.0	476
25	ARC Leros 20	22	293.0	567
26	ARC Leros 20H	22	300.0	408
27	ARC Leros 20R	22	307.0	567
28	MBB Erno CHT 20.0	24	234.0	360
29	EADS CHT 20	25	230.0	395
30	Daimler-Benz CHT 400	400	228.6	325

Table B.4: ADCS uncertainty specifications

Variable	**Probability Distribution**		**Variable**	**Probability Distribution**
A_{max}	$N(5.31, 0.053)$		ψ_{slew6}	$N(8.51, 0.4)$
J_{xx}	$U(300, 450)$		ψ_{slew7}	$N(9.88, 0.5)$
J_{zz}	$U(450, 600)$		ψ_{slew8}	$N(5.64, 0.2)$
R	$N(1.3, 0.0013)$		ψ_{slew9}	$N(5.04, 0.2)$
δ_1	$N(0, 0.5)$		ψ_{slew10}	$N(5.75, 0.2)$
δ_2	$N(0, 0.5)$		ψ_{slew11}	$N(4.47, 0.1)$
g_s	$N(1400, 14)$		ψ_{slew12}	$N(5.53, 0.1)$
κ	$U(0.6, 0.7)$		ψ_{slew13}	$N(5.85, 0.1)$
ω_{spin0}	$N(12, 1.33)$		ψ_{slew14}	$\Gamma(1.5, 10.5)$
ω_{spin1}	$N(2, 0.0667)$		ψ_{slew15}	$\Gamma(1.5, 10.5)$
ω_{spin2}	$\Gamma(11, 0.25)$		ψ_{slew16}	$\Gamma(1.5, 10.5)$
ω_{spin3}	$L(2, 0.0667)$		ψ_{slew17}	$\Gamma(1.5, 10.5)$
ψ_{slew1}	$N(5, 0.5)$		ψ_{slew18}	$\Gamma(1.5, 10.5)$
ψ_{slew2}	$N(50.45, 5)$		ψ_{slew19}	$\Gamma(1.5, 10.5)$
ψ_{slew3}	$N(5.13, 0.5)$		q	$N(0.6, 0.06)$
ψ_{slew4}	$N(6.35, 0.6)$		$uncfuel$	$N(0, 0.05)$
ψ_{slew5}	$N(2.76, 0.2)$		F	$N(F_{table}, 7/300F_{table})$

Index

Bibliography

[1] N.M. Alexandrov and M.Y. Hussaini. Multidisciplinary design optimization: State of the art. In *Proceedings of the ICASE/NASA Langley Workshop on Multidisciplinary Design Optimization*, Hampton, Virginia, USA, 1997.

[2] D.A. Alvarez. On the calculation of the bounds of probability of events using infinite random sets. *International Journal of Approximate Reasoning*, 43(3):241–267, 2006.

[3] A. Aregui and T. Denœux. Constructing predictive belief functions from continuous sample data using confidence bands. In *Proceedings of the 5th International Symposium on Imprecise Probability: Theories and Applications*, pp. 11–19, Prague, Czech Republic, 2007.

[4] A. Bateman, D. Ward, and G. Balas. Robust/worst-case analysis and simulation tools. In *Proceedings of the AIAA Guidance, Navigation, and Control Conference and Exhibit*, San Francisco, California, USA, 2005.

[5] C. Baudrit, A. Helias, and N. Perrot. Uncertainty analysis in food engineering involving imprecision and randomness. In *Proceedings of the 5th International Symposium on Imprecise Probability: Theories and Applications*, pp. 21–30, Prague, Czech Republic, 2007.

[6] T. Bayes. An essay towards solving a problem in the doctrine of chances. *Philosophical Transactions*, 53:370–418, 1763.

[7] M. Beer, M. Liebscher, and B. Möller. Structural design under fuzzy randomness. In *Proceedings of the NSF Workshop on Reliable Engineering Computing*, pp. 215–234, Savannah, Georgia, USA, 2004.

[8] Y. Ben-Haim and I. Elishakoff. *Convex Models of Uncertainty in Applied Mechanics.* Elsevier, 1990.

[9] J.R. Benjamin and C.A. Cornell. *Probability, Statistics, and Decision for Civil Engineers.* McGraw-Hill, New York, 1970.

[10] D. Berleant and H. Cheng. A software tool for automatically verified operations on intervals and probability distributions. *Reliable Computing*, 4(1):71–82, 1998.

[11] D. Berleant, S. Ferson, V. Kreinovich, and W.A. Lodwick. Combining interval and probabilistic uncertainty: Foundations, algorithms, challenges – an overview. In *Proceedings of the 4th International Symposium on Imprecise Probabilities and Their Applications*, Pittsburgh, Pennsylvania, USA, 2005.

[12] D. Berleant and L. Xie. An interval-based tool for verified arithmetic on random variables of unknown dependency. Manuscript, 2005.

[13] A. Bernardini. *Whys and Hows in Uncertainty Modelling: Probability, Fuzziness and Anti-Optimization*, chapter What are random and fuzzy sets and how to use them for uncertainty modelling in engineering systems?, pp. 63–125. Springer, 1999.

[14] J.M. Bernardo and A.F.M. Smith. *Bayesian Theory.* Wiley, 1994.

[15] M. Bickis and U. Bickis. Predicting the next pandemic: An exercise in imprecise hazards. In *Proceedings of the 5th International Symposium on Imprecise Probability: Theories and Applications*, pp. 41–46, Prague, Czech Republic, 2007.

[16] H.H. Bothe. *Fuzzy Logic: Einführung in Theorie und Anwendungen.* Springer, 1993.

[17] A. Bronevich and A. Lepskiy. Measuring uncertainty with imprecision indices. In *Proceedings of the 5th International Symposium on Imprecise Probability: Theories and Applications*, pp. 47–55, Prague, Czech Republic, 2007.

[18] A. Brooke, D. Kendrick, and A. Meeraus. *GAMS: A User's Guide.* The Scientific Press, 1988.

[19] CBDO. Cloud Based Design Optimization webpage `http://www.martin-fuchs.net/cbdo.php`, 2004.

[20] S.H. Chen. Ranking fuzzy numbers with maximizing set and minimizing set. *Fuzzy Sets and Systems*, 17:113–129, 1985.

[21] C.H. Cheng and D.L. Mon. Fuzzy system reliability analysis by interval of confidence. *Fuzzy Sets and Systems*, 56(1):29–35, 1993.

[22] K.L. Chung. *A Course in Probability Theory.* Academic Press, 1968.

[23] F.P.A. Coolen, B. Houlding, and S.G. Parkinson. Jury size and composition – a predictive approach. In *Proceedings of the 5th International Symposium on Imprecise Probability: Theories and Applications*, pp. 87–96, Prague, Czech Republic, 2007.

[24] I. Couso, S. Moral, and P. Walley. Examples of independence for imprecise probabilities. In *Proceedings of the 1st International Symposium on Imprecise Probabilities and Their Applications*, pp. 121–130, Ghent, Belgium, 1999.

[25] F.G. Cozman. Credal networks. *Artificial Intelligence*, 120(2):199–233, 2000.

[26] N. Croisard, M. Ceriotti, and M. Vasile. Uncertainty modelling in reliable preliminary space mission design. In *Workshop on Artificial Intelligence for Space Applications*, Hyderabad, India, 2007.

[27] T. Csendes, L. Pal, J.O.H. Sendin, and J.R. Banga. The GLOBAL optimization method revisited. *Optimization Letters*, 2:445–454, 2008.

[28] A.P. Dempster. Upper and lower probabilities induced by a multivalued mapping. *Annals of Mathematical Statistics*, 38(2):325–339, 1967.

[29] A.P. Dempster. Upper and lower probabilities generated by a random closed interval. *Annals of Mathematical Statistics*, 39(3):957–966, 1968.

[30] T. Denœux. Constructing belief functions from sample data using multinomial confidence regions. *International Journal of Approximate Reasoning*, 42(3):228–252, 2006.

[31] A. Der Kiureghian and T. Dakessian. Multiple design points in first and second-order reliability. *Structural Safety*, 20(1):37–49, 1998.

[32] S. Destercke, D. Dubois, and E. Chojnacki. Relating practical representations of imprecise probabilities. In *Proceedings of the 5th International Symposium on Imprecise Probability: Theories and Applications*, pp. 155–163, Prague, Czech Republic, 2007.

[33] P. Deuflhard. *Newton Methods for Nonlinear Problems: Affine Invariance and Adaptive Algorithms.* Springer, 2004.

[34] D. Dubois and H. Prade. *Possibility Theory: An Approach to Computerized Processing of Uncertainty.* New York: Plenum Press, 1986.

[35] D. Dubois and H. Prade. Interval-valued fuzzy sets, possibility theory and imprecise probability. In *Proceedings of the International Conference in Fuzzy Logic and Technology*, 2005.

[36] EADS. Space Propulsion Web Page
`http://cs.space.eads.net/sp/`, 2007.

[37] M.S. Eldred, S.L. Brown, B.M. Adams, D.M. Dunlavy, D.M. Gay, L.P. Swiler, A.A. Giunta, W.E. Hart, J.-P. Watson, J.P. Eddy, J.D. Griffin, P.D. Hough, T.G. Kolda, M.L. Martinez-Canales, and P.J. Williams. DAKOTA, A Multilevel Parallel Object-Oriented Framework for Design Optimization, Parameter Estimation, Uncertainty Quantification, and Sensitivity Analysis: Version 4.0 Users Manual. Sand Report SAND2006-6337, Sandia National Laboratories, 2006.

[38] I. Elishakoff. *Whys and Hows in Uncertainty Modelling: Probability, Fuzziness and Anti-Optimization*, chapter Are probabilistic and anti-optimization approaches compatible?, pp. 263–355. Springer, 1999.

[39] I. Elishakoff. *Whys and Hows in Uncertainty Modelling: Probability, Fuzziness and Anti-Optimization*, chapter What may go wrong with probabilistic methods?, pp. 265–283. Springer, 1999.

[40] I. Elishakoff. *Whys and Hows in Uncertainty Modelling: Probability, Fuzziness and Anti-Optimization*, chapter Are probabilistic and anti-optimization methods interrelated?, pp. 285–317. Springer, 1999.

[41] I. Elishakoff and Qiang Li. *Whys and Hows in Uncertainty Modelling: Probability, Fuzziness and Anti-Optimization*, chapter How to combine probabilistic and anti-optimization methods?, pp. 319–339. Springer, 1999.

[42] J.K. Erickson. Mars Exploration Rover: Launch, cruise, entry, descent, and landing. In *55th International Astronautical Congress of the International Astronautical Federation, the International Academy of Astronautics, and the International Institute of Space Law*, Vancouver, Canada, October 2004.

[43] ESA CDF. X-ray Evolving Universe Spectroscopy. CDF Study report CDF-31(A), European Space Agency, 2004.

[44] S. Ferson. What Monte Carlo methods cannot do. *Human and Ecological Risk Assessment*, 2:990–1007, 1996.

[45] S. Ferson. *Ramas Risk Calc 4.0 Software: Risk Assessment with Uncertain Numbers.* Lewis Publishers, U.S., 2002.

[46] S. Ferson, L. Ginzburg, and R. Akcakaya. Whereof one cannot speak: When input distributions are unknown. *Risk Analysis*, 1996. In press, available on-line at: `http://www.ramas.com/whereof.pdf`.

[47] S. Ferson, L. Ginzburg, V. Kreinovich, L. Longpre, and M. Aviles. Exact bounds on finite populations of interval data. *Reliable Computing*, 11 (3):207–233, 2005.

[48] S. Ferson, L. Ginzburg, V. Kreinovich, and J. Lopez. Absolute bounds on the mean of sum, product, etc.: A probabilistic extension of interval arithmetic. In *Extended Abstracts of the 2002 SIAM Workshop on Validated Computing*, pp. 70–72, Toronto, Canada, 2002.

[49] S. Ferson, V. Kreinovich, L. Ginzburg, D.S. Myers, and K. Sentz. Constructing probability boxes and Dempster-Shafer structures. Sand Report SAND2002-4015, Sandia National Laboratories, 2003. Available on-line at
`http://www.sandia.gov/epistemic/Reports/SAND2002-4015.pdf`.

[50] T. Fetz. Finite element method with fuzzy parameters. In *Proceedings of the IMACS Symposium on Mathematical Modelling*, vol. 11, pp. 81–86, Vienna, Austria, 1997.

[51] T. Fetz. Sets of joint probability measures generated by weighted marginal focal sets. In *Proceedings of the 2nd International Symposium on Imprecise Probabilities and Their Applications*, pp. 171–178, Maastricht, The Netherlands, 2001.

[52] T. Fetz, M. Oberguggenberger, J. Jager, D. Koll, G. Krenn, H. Lessmann, and R.F. Stark. Fuzzy models in geotechnical engineering and construction management. *Computer-Aided Civil and Infrastructure Engineering*, 14(2):93–106, 1999.

[53] T. Fetz, M. Oberguggenberger, and S. Pittschmann. Applications of possibility and evidence theory in civil engineering. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 8(3):295–309, 2000.

[54] P.I. Fierens. An extension of chaotic probability models to real-valued variables. In *Proceedings of the 5th International Symposium on Imprecise Probability: Theories and Applications*, pp. 193–201, Prague, Czech Republic, 2007.

[55] R.S. Filho, F.G. Cozman, F.W. Trevizan, C.P. de Campos, and L.N. de Barros. Multilinear and integer programming for Markov decision processes with imprecise probabilities. In *Proceedings of the 5th International Symposium on Imprecise Probability: Theories and Applications*, pp. 395–403, Prague, Czech Republic, 2007.

[56] R. Fourer, D.M. Gay, and B.W. Kernighan. *AMPL: A Modeling Language for Mathematical Programming.* Duxbury Press / Brooks/Cole Publishing Company, 2002.

[57] F.N. Fritsch and R.E. Carlson. Monotone piecewise cubic interpolation. *SIAM Journal on Numerical Analysis*, 17(2):238–246, 1980.

[58] M. Fuchs, D. Girimonte, D. Izzo, and A. Neumaier. *Robust Intelligent Systems*, chapter Robust and automated space system design. Springer, 2008. In press, preprint available on-line at:
`http://www.martin-fuchs.net/publications.php`.

[59] M. Fuchs and A. Neumaier. Autonomous robust design optimization with potential clouds. *International Journal of Reliability and Safety, Special Issue on Reliable Engineering Computing*, 2008. Accepted, preprint available on-line at:
`http://www.martin-fuchs.net/publications.php`.

[60] M. Fuchs and A. Neumaier. Potential based clouds in robust design optimization. *Journal of Statistical Theory and Practice, Special Issue*

on Imprecision, 2008. Accepted, preprint available on-line at: `http://www.martin-fuchs.net/publications.php`.

[61] I.J. Good. Subjective probability as the measure of a non-measurable set. *Logic, Methodology and Philosophy of Science*, 1, 1962.

[62] W.T. Grandy and L.H. Schick. *Maximum Entropy and Bayesian Methods*. Springer, 1990.

[63] M.C. Grant and S.P. Boyd. CVX: A system for disciplined convex programming.
`http://www.stanford.edu/~boyd/cvx/cvx_usrguide.pdf`
`http://www.stanford.edu/~boyd/cvx/`, 2007.

[64] R. Haenni. Climbing the hills of compiled credal networks. In *Proceedings of the 5th International Symposium on Imprecise Probability: Theories and Applications*, pp. 213–221, Prague, Czech Republic, 2007.

[65] N. Hansen and A. Ostermeier. Completely derandomized self-adaptation in evolution strategies. *Evolutionary Computation*, 9(2):159–195, 2001.

[66] W.K. Hastings. Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*, 57(1):97–109, 1970.

[67] F. Hofbauer. Maß- und Integrationstheorie. Manuscript, 2005. Available on-line at: `http://www.mat.univie.ac.at/~fh/math.pdf`.

[68] M. Hohenbichler and R. Rackwitz. Non-normal dependent vectors in structural safety. *Journal of the Engineering Mechanics Division*, 107 (6):1227–1238, 1981.

[69] M. Hohenbichler and R. Rackwitz. Improvement of second-order reliability estimates by importance sampling. *Journal of Engineering Mechanics*, 114(12):2195–2199, 1988.

[70] F. Höppner, F. Klawonn, R. Kruse, and T. Runkler. *Fuzzy Cluster Analysis: Methods for Classification, Data Analysis and Image Recognition*. Wiley, 1999.

[71] W. Huyer and A. Neumaier. Global optimization by multilevel coordinate search. *Journal of Global Optimization*, 14(4):331–355, 1999.

[72] W. Huyer and A. Neumaier. SNOBFIT – Stable Noisy Optimization by Branch and Fit. *ACM Transactions on Mathematical Software*, 35(2), 2008. Article 9, 25 pages.

[73] R. Jain. Decision making in the presence of fuzzy variables. *IEEE Transactions on Systems, Man, and Cybernetics*, 6(10):698–703, 1976.

[74] K.D. Jamison and W.A. Lodwic. The construction of consistent possibility and necessity measures. UCD/CCM Report 171, Center for Computational Mathematics, University of Colorado, Denver, USA, 2001.

[75] R. Jirousek, J. Vejnarova, and M. Daniel. Compositional models of belief functions. In *Proceedings of the 5th International Symposium on Imprecise Probability: Theories and Applications*, pp. 243–251, Prague, Czech Republic, 2007.

[76] T. Kapitaniak. *Chaos for Engineers: Theory, Applications, and Control.* Springer, 2000.

[77] L. Kaufman and P.J. Rousseeuw. *Finding Groups in Data: An Introduction to Cluster Analysis.* Wiley, 1990.

[78] I. Kaymaz and K. Marti. Reliability-based design optimization for elastoplastic mechanical structures. *Computers & Structures*, 85(10):615–625, 2007.

[79] W. Kijawatworawet, H.J. Pradlwarter, and G.I. Schueller. Structural reliability estimation by adaptive importance directional sampling. In *Structural Safety and Reliability: Proceedings of ICOSSAR'97*, pp. 891–897, Balkema, 1998.

[80] P.N. Koch, T.W. Simpson, J.K. Allen, and F. Mistree. Statistical approximations for multidisciplinary optimization: The problem of size. *Special Issue on Multidisciplinary Design Optimization of Journal of Aircraft*, 36(1):275–286, 1999.

[81] A. Kolmogorov. *Grundbegriffe der Wahrscheinlichkeitsrechnung.* Springer, Berlin, 1933.

[82] A. Kolmogorov. Confidence limits for an unknown distribution function. *The Annals of Mathematical Statistics*, 12(4):461–463, 1941.

[83] O.M. Kosheleva and M. Ceberio. Processing educational data: From traditional statistical techniques to an appropriate combination of probabilistic, interval, and fuzzy approaches. In *Proceedings of the International Conference on Fuzzy Systems, Neural Networks, and Genetic Algorithms*, pp. 39–48, Tijuana, Mexico, 2005.

[84] I. Kozine and V. Krymsky. Enhancement of natural extension. In *Proceedings of the 5th International Symposium on Imprecise Probability: Theories and Applications*, pp. 253–261, Prague, Czech Republic, 2007.

[85] V. Krätschmer. A unified approach to fuzzy random variables. *Fuzzy Sets and Systems*, 123(1):1–9, 2001.

[86] V. Kreinovich. *Random Sets: Theory and Applications*, chapter Random sets unify, explain, and aid known uncertainty methods in expert systems, pp. 321–345. Springer, 1997.

[87] V. Kreinovich. Probabilities, intervals, what next? optimization problems related to extension of interval computations to situations with partial information about probabilities. *Journal of Global Optimization*, 29(3):265–280, 2004.

[88] V. Kreinovich, J. Beck, C. Ferregut, A. Sanchez, G.R. Keller, M. Averill, and S.A. Starks. Monte-Carlo-type techniques for processing interval uncertainty, and their engineering applications. In *Proceedings of the NSF Workshop on Reliable Engineering Computing*, pp. 139–160, Savannah, Georgia, USA, 2004.

[89] V. Kreinovich and S. Ferson. A new Cauchy-based black-box technique for uncertainty in risk analysis. *Reliability Engineering & System Safety*, 85(1–3):267–279, 2004.

[90] V. Kreinovich, S. Ferson, and L. Ginzburg. Exact upper bound on the mean of the product of many random variables with known expectations. *Reliable Computing*, 9(6):441–463, 2003.

[91] V. Kreinovich and R. Trejo. *Handbook of Randomized Computing*, chapter Error estimations for indirect measurements: randomized vs. deterministic algorithms for 'black-box' programs, pp. 673–729. Kluwer, 2001.

[92] H. Kwakernaak. Fuzzy random variables – ii: Algorithms and examples for the discrete case. *Information Sciences*, 17:253–278, 1979.

[93] W.J. Larson and J.R. Wertz. *Space Mission Analysis and Design.* Microcosm Press, 3 ed., 1999.

[94] W.A. Lodwick and K.D. Jamison. Estimating and validating the cumulative distribution of a function of random variables: Toward the development of distribution arithmetic. *Reliable Computing*, 9(2):127–141, 2003.

[95] K. Makino and M. Berz. Efficient control of the dependency problem based on taylor model methods. *Reliable Computing*, 5(1):3–12, 1999.

[96] K. Marti and I. Kaymaz. Reliability analysis for elastoplastic mechanical structures under stochastic uncertainty. *Zeitschrift fr Angewandte Mathematik und Mechanik*, 86(5):358–384, 2006.

[97] D.J. McCormick and J.R. Olds. A distributed framework for probabilistic analysis. In *AIAA/ISSMO Symposium On Multidisciplinary Analysis And Design Optimization*, Atlanta, Georgia, USA, 2002.

[98] M.D. McKay, W.J. Conover, and R.J. Beckman. A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. *Technometrics*, 221:239–245, 1979.

[99] MER. Mars Exploration Rover Project
`http://marsrovers.nasa.gov/mission/spacecraft.html`, 2003.

[100] N. Metropolis, A.W. Rosenbluth, M.N. Rosenbluth, A.H. Teller, and E. Teller. Equations of state calculations by fast computing machines. *Journal of Chemical Physics*, 21(6):1087–1092, 1953.

[101] I. Molchanov. *Theory of Random Sets.* Springer, 2005.

[102] B. Möller and M. Beer. *Fuzzy Randomness: Uncertainty in Civil Engineering and Computational Mechanics.* Springer-Verlag Berlin Heidelberg, 2004.

[103] B. Möller, M. Beer, W. Graf, and J. U. Sickert. Fuzzy finite element method and its application. In *Trends in Computational Structural Mechanics*, pp. 529–538, Barcelona, Spain, 2001.

[104] B. Möller, M. Beer, and U. Reuter. Theoretical basics of fuzzy-randomness – application to time series with fuzzy data. In *Safety*

and Reliability of Engineering Systems and Structures: Proceedings of the 9th International Conference on Structural Safety and Reliability, Rome, Italy, 2005.

[105] B. Möller, W. Graf, and M. Beer. Fuzzy structural analysis using α-level optimization. *Computational Mechanics*, 26(6):547–565, 2000.

[106] B. Möller, W. Graf, and M. Beer. Fuzzy probabilistic method and its application for the safety assessment of structures. In *Proceedings of the European Conference on Computational Mechanics*, Cracow, Poland, 2001.

[107] R.E. Moore. *Methods and Applications of Interval Analysis.* Society for Industrial & Applied Mathematics, 1979.

[108] Z.P. Mourelatos and J. Liang. An efficient unified approach for reliability and robustness in engineering design. In *Proceedings of the NSF Workshop on Reliable Engineering Computing*, pp. 119–138, Savannah, Georgia, USA, 2004.

[109] Z.P. Mourelatos and J. Zhou. Reliability estimation and design with insufficient data based on possibility theory. *AIAA Journal*, 43(8):1696–1705, 2005.

[110] Z.P. Mourelatos and J. Zhou. A design optimization method using evidence theory. *Journal of Mechanical Design*, 128(4):901–908, 2006.

[111] R.L. Muhanna and R.L. Mullen. Formulation of fuzzy finite element methods for mechanics problems. *Computer-Aided Civil and Infrastructure Engineering*, 14(2):107–117, 1999.

[112] R.L. Muhanna and R.L. Mullen. Uncertainty in mechanics problems – interval-based approach. *Journal of Engineering Mechanics*, 127(6): 557–566, 2001.

[113] A. Nataf. Determination des distributions de probabilites dont les marges sont donnees. *Comptes Rendus de lAcademie des Sciences*, 225: 42–43, 1962.

[114] NEOS. NEOS Server for Optimization
`http://www-neos.mcs.anl.gov/`, 2008.

[115] A. Neumaier. *Interval Methods for Systems of Equations.* Cambridge University Press, 1990.

[116] A. Neumaier. Clouds, fuzzy sets and probability intervals. *Reliable Computing*, 10(4):249–272, 2004. Available on-line at: `http://www.mat.univie.ac.at/~neum/ms/cloud.pdf`.

[117] A. Neumaier. Uncertainty modeling for robust verifiable design. Slides, 2004. Available on-line at: `http://www.mat.univie.ac.at/~neum/ms/uncslides.pdf`.

[118] A. Neumaier, M. Fuchs, E. Dolejsi, T. Csendes, J. Dombi, B. Banhelyi, and Z. Gera. Application of clouds for modeling uncertainties in robust space system design. ACT Ariadna Research ACT-RPT-05-5201, European Space Agency, 2007. Available on-line at `http://www.esa.int/gsp/ACT/ariadna/completed.htm`.

[119] H. Nguyen. Fuzzy sets and probability. In *Proceedings of the 6th International Fuzzy Systems Association World Congress*, vol. 1, pp. 11–13, Brasil, 1995.

[120] H. Nguyen. Fuzzy sets and probability. *Fuzzy Sets and Systems, Secial Issue on Fuzzy Sets: Where do we stand? Where do we go?*, 90(2): 129–132, 1997.

[121] Y. Noh, K.K. Choi, and L. Du. New transformation of dependent input variables using copula for RBDO. In *7th World Congress on Structural and Multidisciplinary Optimization*, Seoul, Korea, 2007.

[122] U. Nowak and L. Weimann. A Family of Newton Codes for Systems of Highly Nonlinear Equations – Algorithm, Implementation, Application. Technical report, Konrad-Zuse-Zentrum fuer Informationstechnik Berlin, 1990.

[123] M. Oberguggenberger and W. Fellin. Assessing the sensitivity of failure probabilities: a random set approach. In *Safety and Reliability of Engineering Systems and Structures: Proceedings of the 9th International Conference on Structural Safety and Reliability*, pp. 1755–1760, Rome, Italy, 2005.

[124] M. Oberguggenberger and W. Fellin. Reliability bounds through random sets: Non-parametric methods and geotechnical applications. *Computers and Structures*, 86(10):1093–1101, 2008.

[125] M. Oberguggenberger, J. King, and B. Schmelzer. Imprecise probability methods for sensitivity analysis in engineering. In *Proceedings of the 5th International Symposium on Imprecise Probability: Theories and Applications*, pp. 317–325, Prague, Czech Republic, 2007.

[126] M. Papadrakakis and N.D. Lagaros. Reliability-based structural optimization using neural networks and Monte Carlo simulation. *Computer Methods in Applied Mechanics and Engineering*, 191(32):3491–3507, 2002.

[127] M. Pate-Cornell and P. Fischbeck. Probabilistic risk analysis and risk based priority scale for the tiles of the space shuttle. *Reliability Engineering and System Safety*, 40(3):221–238, 1993.

[128] Personal communication with ESA engineers, 2007.

[129] A. Pownuk. Calculation of displacement in elastic and elastic-plastic structures with interval parameters. In *Proceedings of the 33rd Solid Mechanics Conference*, pp. 5–9, Zakopane, Poland, 2000.

[130] A. Pownuk. General interval FEM program based on sensitivity analysis method. In *Proceedings of the 3rd International Workshop on Reliable Engineering Computing*, pp. 397–428, Savannah, Georgia, USA, 2008.

[131] W.H. Press, S.A. Teukolsky, W.T. Vetterling, and B.P. Flannery. *Numerical Recipes in C*. Cambridge University Press, 2nd ed., 1992.

[132] Purdue School of Aeronautics and Astronautics. Satellite Propulsion Web Page, `http://cobweb.ecn.purdue.edu/~propulsi/propulsion/`, 1998.

[133] M. Puri and D. Ralescu. *Madan Lal Puri Selected Collected Works, Volume 3: Time Series, Fuzzy Analysis and Miscellaneous Topics*, chapter Fuzzy Random Variables. Brill Academic Pub, 2003.

[134] R. Rackwitz. Reliability analysis – a review and some perspectives. *Structural Safety*, 23(4):365–395, 2001.

[135] L.C. Rego. Conditioning in chaotic probabilities interpreted as a generalized Markov chain. In *Proceedings of the 5th International Symposium on Imprecise Probability: Theories and Applications*, pp. 365–373, Prague, Czech Republic, 2007.

[136] A. Richards, J. How, T. Schouwenaars, and E. Feron. Plume avoidance maneuver planning using mixed integer linear programming. In *Proceedings of the AIAA Guidance, Navigation, and Control Conference and Exhibit*, Montreal, Canada, 2001.

[137] M. Rosenblatt. Remarks on a multivariate transformation. *Annals of Mathematical Statistics*, 23(3):470–472, 1952.

[138] T.J. Ross. *Fuzzy Logic with Engineering Applications.* New York, NY: McGraw-Hill, 1995.

[139] E. Saad. Structural optimization based on evolution strategy. In *Advanced Computational Methods in Structural Mechanics*, pp. 266–280, Barcelona, Spain, 1996.

[140] E. Saad. Qualitative and quantitative reasoning in hybrid probabilistic logic programs. In *Proceedings of the 5th International Symposium on Imprecise Probability: Theories and Applications*, pp. 375–383, Prague, Czech Republic, 2007.

[141] L. Schrage. *Optimization Modeling with LINGO.* LINDO Systems, 2008.

[142] G.I. Schueller. A state-of-the-art report on computational stochastic mechanics. *Probabilistic Engineering Mechanics*, 12(4):197–321, 1997.

[143] R.G. Sexsmith. Probability-based safety analysis – value and drawbacks. *Structural Safety*, 21(4):303–310, 1999.

[144] G. Shafer. *A Mathematical Theory of Evidence.* Princeton University Press, 1976.

[145] C.E. Shannon and W. Weaver. *The Mathematical Theory of Communication.* University of Illinois Press, 1949.

[146] J.U. Sickert, W. Graf, and U. Reuter. Application of fuzzy randomness to time-dependent reliability. In *Safety and Reliability of Engineering Systems and Structures: Proceedings of the 9th International Conference on Structural Safety and Reliability*, pp. 1709–1716, Rome, Italy, 2005.

[147] A. Sklar. Fonctions de repartition a n dimensions et leurs marges. *Publications de l'Institut de Statistique de L'Universite de Paris*, 8:229–231, 1959.

[148] D. Skulj. *Soft Methods for Integrated Uncertainty Modelling*, chapter Finite Discrete Time Markov Chains with Interval Probabilities, pp. 299–306. Springer, 2006.

[149] D. Skulj. Regular finite Markov chains with interval probabilities. In *Proceedings of the 5th International Symposium on Imprecise Probability: Theories and Applications*, pp. 405–413, Prague, Czech Republic, 2007.

[150] B.A. Smith, S.P. Kenny, and L.G. Crespo. Probabilistic parameter uncertainty analysis of single input single output control systems. NASA Technical Report NASA/TM-2005-213280, NASA, 2005.

[151] C.A.B. Smith. Consistency in statistical inference and decision. *Journal of the Royal Statistical Society*, 23(1):1–37, 1961.

[152] S.A. Starks, V. Kreinovich, L. Longpre, M. Ceberio, G. Xiang, R. Araiza, J. Beck, R. Kandathi, A. Nayak, and R. Torres. Towards combining probabilistic and interval uncertainty in engineering calculations. In *Proceedings of the NSF Workshop on Reliable Engineering Computing*, pp. 193–214, Savannah, Georgia, USA, 2004.

[153] M. Sugeno. *Theory of fuzzy integrals and its applications.* PhD thesis, Tokyo Institute of Technology, 1974.

[154] M. Sugeno. An introductory survey of fuzzy control. *Information Science*, 36:59–83, 1985.

[155] E.J.W. ter Marten, T.S. Doorn, J.A. Croon, A. Bargagli, A. di Bucchianico, and O. Wittich. Importance sampling for high speed statistical Monte-Carlo simulations. NXP Technical Note NXP-TN-2007-00238, NXP Semiconductors, 2007.

[156] D.P. Thunnissen. *Propagating and Mitigating Uncertainty in the Design of Complex Multidisciplinary Systems.* PhD thesis, California Institute of Technology Pasadena, 2005.

[157] L. Utkin and S. Destercke. Computing expectations with p-boxes: two views of the same problem. In *Proceedings of the 5th International Symposium on Imprecise Probability: Theories and Applications*, pp. 435–443, Prague, Czech Republic, 2007.

[158] R. Viertl. *Statistical Methods for Non-Precise Data.* CRC Press, 1996.

[159] P. Walley. *Statistical Reasoning with Imprecise Probability.* Chapman and Hall, 1991.

[160] P. Walley. Measures of uncertainty in expert systems. *Artificial Intelligence*, 83(1):1–58, 1996.

[161] P. Walley. Towards a unified theory of imprecise probability. *International Journal of Approximate Reasoning*, 24(2–3):125–148, 2000.

[162] Z. Wang and G.J. Klir. *Fuzzy Measure Theory.* Plenum Publishing Corporation, 1992.

[163] K. Weichselberger. The theory of interval-probability as a unifying concept for uncertainty. *International Journal of Approximate Reasoning*, 24(2–3):149–170, 2000.

[164] E.W. Weisstein. Cauchy distribution. MathWorld – A Wolfram Web Resource, 2008. Available on-line at: `http://mathworld.wolfram.com/CauchyDistribution.html`.

[165] P. Whittle. *Probability via Expectation.* Springer, 1992.

[166] R.C. Williamson. *Probabilistic Arithmetic.* PhD thesis, University of Queensland, 1989.

[167] R.C. Williamson and T. Downs. Probabilistic arithmetic. I. numerical methods for calculating convolutions and dependency bounds. *International Journal of Approximate Reasoning*, 4(2):89–158, 1990.

[168] XEUS. X-ray Evolving Universe Spectroscopy `http://www.esa.int/esaSC/120369_index_0_m.html`, 2004.

[169] L.A. Zadeh. Fuzzy sets. *Information and Control*, 8(3):338–353, 1965.

[170] L.A. Zadeh. Similarity relations and fuzzy orderings. *Information Sciences*, 3(2):177–200, 1971.

[171] L.A. Zadeh. Fuzzy sets as a basis for a theory of possibility. *Fuzzy Sets and Systems*, 1:3–28, 1978.

[172] J. Zhou and Z.P. Mourelatos. Design under uncertainty using a combination of evidence theory and a Bayesian approach. In *Proceedings of the 3rd International Workshop on Reliable Engineering Computing*, pp. 171–198, Savannah, Georgia, USA, 2008.

[173] H.J. Zimmermann. *Fuzzy Set Theory – and Its Applications.* Kluwer Academic Publishers, 2001.

[174] A. Zonca. Modelling and Optimisation of Space Mission Prephase A Design Process in a Concurrent Engineering Environment Through a Decision-making Software Based on Expert Systems Theory. Manuscript, 2004.

Uncertainty modeling in higher dimensions: Towards robust design optimization

Deutsche Zusammenfassung

Moderne Design Probleme stellen Ingenieure vor mehrere elementare Aufgaben.

- Das Design muss die angestrebten Funktionalitäten aufweisen.
- Es muss optimal sein in Hinsicht auf eine vorgegebene Zielfunktion.
- Schließlich muss das Design abgesichert sein gegen Unsicherheiten, die nicht zu Versagen des Designs führen dürfen.

All diese Aufgaben lassen sich unter dem Begriff der robusten Design Optimierung zusammenfassen und verlangen nach computergestützten Methoden, die Unsicherheitsmodellierung und Design Optimierung in sich vereinen.

Unsicherheitsmodellierung enthält einige fundamentale Herausforderungen: Der Rechenaufwand darf gewisse Grenzen nicht überschreiten; unbegründete Annahmen müssen so weit wie möglich vermieden werden. Die beiden kritischsten Probleme betreffen allerdings den Umgang mit unvollständiger stochastischer Information und mit hoher Dimensionalität. Der niedrigdimensionale Fall ist gut erforscht, und es existieren diverse Methoden, auch unvollständige Informationen zu verarbeiten. In höheren Dimensionen hingegen ist die Anzahl der Möglichkeiten derzeit sehr begrenzt. Ungenauigkeit und Unvollständigkeit von Daten kann schwerwiegende Probleme verursachen – aber die Lage ist nicht hoffnungslos.

In dieser Dissertation zeigen wir, wie man den hochdimensionalen Fall mit Hilfe von "Potential Clouds" in ein eindimensionales Problem übersetzt. Dieser Ansatz führt zu einer Unsicherheitsanalyse auf Konfidenzregionen relevanter Szenarien mittels einer Potential Funktion. Die Konfidenzregionen werden als Nebenbedingungen in einem Design Optimierungsproblem formuliert. Auf diese Weise verknüpfen wir Unsicherheitsmodellierung und Design Optimierung, wobei wir außerdem eine adaptive Aktualisierung der Unsicherheitsinformationen ermöglichen. Abschließend wenden wir unsere Methode in zwei Fallstudien an, in 24, bzw. in 34 Dimensionen.

Vita

Full Name:	Martin Ernst Fuchs
Date of Birth:	23/07/1981
Place of Birth:	Duisburg, Germany
Citizenship:	German

2005-Present:

- Ph.D. studies in Mathematics at the University of Vienna, Faculty of Mathematics

2006-Present:

- Assistent in Ausbildung at the University of Vienna, Faculty of Mathematics, including teaching (Proseminar zu Lineare Algebra und Geometrie 2)

2008:

- Presentation on 'Potential clouds in robust design' at the CSCS '08, Szeged, Hungary, 02-05/07/08
- Presentation on 'Uncertainty modeling with clouds in autonomous robust design optimization' at the REC '08, Savannah, Georgia, USA, 20-22/02/08

2007:

- Invited 4 months stage at the European Space Agency, Advanced Concepts Team, Noordwijk, The Netherlands
- Presentation on 'Uncertainty modeling in autonomous robust spacecraft system design' at the ICIAM '07, Zürich, Switzerland, 16-20/07/07

2006:

- Wissenschaftlicher Mitarbeiter at the University of Vienna, Faculty of Mathematics

2000-2005:

- Mag.rer.nat (M.Sc.)
- Studies of Mathematics at the University of Vienna, Faculty of Mathematics, finished with distinction on 13/01/2005

For a full list of my papers, preprints, abstracts, and slides please visit my website `martin-fuchs.net`.