Uncertainty modeling with clouds in autonomous robust design optimization

Martin Fuchs and Arnold Neumaier

University of Vienna
Faculty of Mathematics
Nordbergstr. 15
1090 Wien
Austria
email: martin.fuchs@univie.ac.at

Abstract. The task of autonomous and robust design cannot be regarded as a single task, but consists of two tasks that have to be accomplished concurrently. First, the design should be found autonomously; this indicates the existence of a method which is able to find the optimal design choice automatically. Second, the design should be robust; in other words: the design should be safeguarded against uncertain perturbations.

Traditional modeling of uncertainties faces several problems. The lack of knowledge about distributions of uncertain variables or about correlations between uncertain data, respectively, typically leads to underestimation of error probabilities. Moreover, in higher dimensions the numerical computation of the error probabilities is very expensive, if not impossible, even provided the knowledge of the multivariate probability distributions.

Based on the *clouds* formalism we have developed new methodologies to gather all available uncertainty information from expert engineers, process it to a reliable worst-case analysis and finally optimize the design seeking the optimal robust design.

The new methods are applied to problems for autonomous optimization in robust spacecraft system design at the European Space Agency (ESA).

Keywords: uncertainty modeling, robust design, clouds, autonomous design, design optimization

Acknowledgements

I would like to acknowledge the support of the people from the ESA Advanced Concepts Team who contributed to this paper in various ways, in particular I would like to thank Daniela Girimonte and Dario Izzo.

1. Introduction

In general terms, uncertainty handling for design optimization has the goal to safeguard reliably against uncertain perturbations while seeking an optimal design. The achieved design can thus be qualified as robust.

An engineer who designs a structure faces the task to develop a product which satisfies given requirements formulated as design constraints. Output of the engineer's work should be an optimal design with respect to a certain design objective. In many cases this is the cost or the mass of the designed product. An algorithmic method for design optimization functions as decision making support for engineers. In the last years, much research has been dedicated to the achievement of decisions support systems. Even the attempt of autonomous design has been made trying to capture the reasoning of the system experts. For more complex kinds of structures, e.g., a spacecraft component or a whole spacecraft, the design process involves several different engineering fields, so the design optimization becomes multidisciplinary, and an interaction between the comprised disciplines is necessary. The resulting overall optimization process is known as multidisciplinary design optimization (MDO). Design related uncertainties are handled to safeguard against failures of the design, i.e., a violation of the design requirement constraints, caused by uncertain errors.

In many cases, in particular for early design phases, it is common engineering practice to handle uncertainties by assigning intervals, or safety margins, to the uncertain variables, usually combined with an iterative process of refining the intervals while converging to a robust optimal design. The refinement of the intervals is done by experts who assess whether the worst-case scenario, that has been determined for the design at the current stage of the iteration process, is too pessimistic or too optimistic. How to assign the intervals and how to choose the endpoint of the assigned intervals to get the worst-case scenario is usually not computed but assessed by an expert. The goal of the whole iteration includes both optimization of the design and safeguarding against uncertainties. Apart from interval assignments there are further ways to handle uncertainties in design processes, e.g., methods from probability theory or fuzzy theory like fuzzy clustering, portfolio theory, or simulation techniques like Monte Carlo.

Real life applications of uncertainty methods disclose various problems. The dimension of many uncertain real life scenarios is very high which causes severe computational problems, famous as the curse of dimensionality, see, e.g., (Koch et al., 1999). Even given the knowledge of the multivariate probability distributions the numerical computation of the error probabilities becomes very expensive, if not impossible. Moreover, the available uncertainty information in early design phases is often very limited, mostly there are only interval bounds on the uncertain variables, sometimes probability distributions for single variables without correlation information. When the amount of uncertainty information available is small, traditional methods face additional problems. To make use of well-known current methods from probability or fuzzy theory more such information would be required. Simulation techniques also require a larger amount of information to be reliable, or unjustified assumptions on the uncertainties have to be made. The lack of information typically causes these methods to underestimate the effects of the uncertain tails of the probability distribution, cf. (Ferson, 1996). Similarly, a reduction of the problem to an interval analysis after assigning intervals to the uncertain variables as described before (e.g., 3 σ boxes) entails a loss of valuable uncertainty information which would actually be available, maybe unformalized, but is not at all involved in the uncertainty model.

Many previous works are dedicated to MDO or robust design. In a classical approach to MDO, cf. (Alexandrov and Hussaini, 1997), (Roy, 1996), (Belton and Stewart, 2002), each specialist would prepare a subsystem design rather independently, using stand-alone tools. Design iterations among the different discipline experts would take place in meetings at certain time intervals. This well-established approach reduces the opportunity to find interdisciplinary solutions and to create system awareness in the specialists. A considerable step forward in MDO for early design phases has been achieved by concurrent engineering where a sequential iterative routine is replaced by a parallel and cooperative procedure. Facilities where these methodologies are implemented for the special case of spacecraft design are, among others, the ESA Concurrent Design Facility (Bandecchi et al., 1999), the NASA Goddard Integrated Mission Design Center (Karpati et al., 2003) and the Concept Design Center at The AeroSpace Corporation (Aguilar et al., 1998). An approach to MDO via game theory can be found, e.g., in (Lewis and Mistree, 1997). To improve the robustness in the process of design optimization there are various approaches dealing with uncertainty modeling. In (Pate-Cornell and Fischbeck, 1993) probability risk analysis is applied to the uncertainties in space shuttle design; an approach from fuzzy theory can be found, e.g., in (Ross, 1995); in (Thunnissen, 2005) a general qualitative and quantitative investigation of uncertainties in space design is given. The work by (Amata et al., 2004) presents studies harmonizing the interests from different disciplines in multidisciplinary design optimization. The attempt to incorporate both uncertainty and autonomy in the design process was made, e.g., in (McCormick and Olds, 2002), using Monte-Carlo simulation techniques, or in (Lavagna and Finzi, 2002), with a fuzzy logic approach.

The ESA Advanced Concepts Team in cooperation with the University of Vienna performed an Ariadna study on the application of the clouds theory in space design optimization, cf. (Neumaier et al., 2007). This study presented an initial step on how clouds could be applied to handle uncertainties in spacecraft design. A significant further step is given in (Fuchs et al., 2007).

Deepening the understanding of the latter studies, we here focus on the theory of clouds and emphasize the capability of an adaptive processing of unformalized uncertainty information with clouds. Clouds allow the representation of incomplete stochastic information in a clearly understandable and computationally attractive way, mediating between aspects of fuzzy set theory and probability distributions, cf. (Dubois and Prade, 2005). The use of clouds permits an adaptive worst-case analysis without losing track of important probabilistic information. At the same time, all computed probabilities, and hence the resulting designs, are reasonably safeguarded against perturbations due to unmodeled and possibly unavailable information. For given confidence levels, the clouds provide regions of relevant scenarios affecting the worst-case for a given design. We have the ambitious goal to achieve a quantification of reliability close to classical probability theory methods, but in higher dimensional spaces of uncertain scenarios so that we can deal with real-life design problems. To find a reliable robust and optimal design autonomously, we have additionally developed heuristic optimization methods.

Figure 1 illustrates the basic concept of our approach. The expert provides the underlying model, given as a black-box model, and all currently available uncertainty information on the model inputs. The information is processed to generate a cloud that provides a nested collection of regions of relevant scenarios parameterized by a confidence level α, and thus produces safety constraints for the optimization. The optimization minimizes a certain objective function (e.g., cost, mass) subject to the safety constraints and to the functional constraints which are represented by the underlying model. The results of the optimization are returned to the expert, who is given an interactive possibility to provide additional uncertainty information afterwards and rerun the procedure.

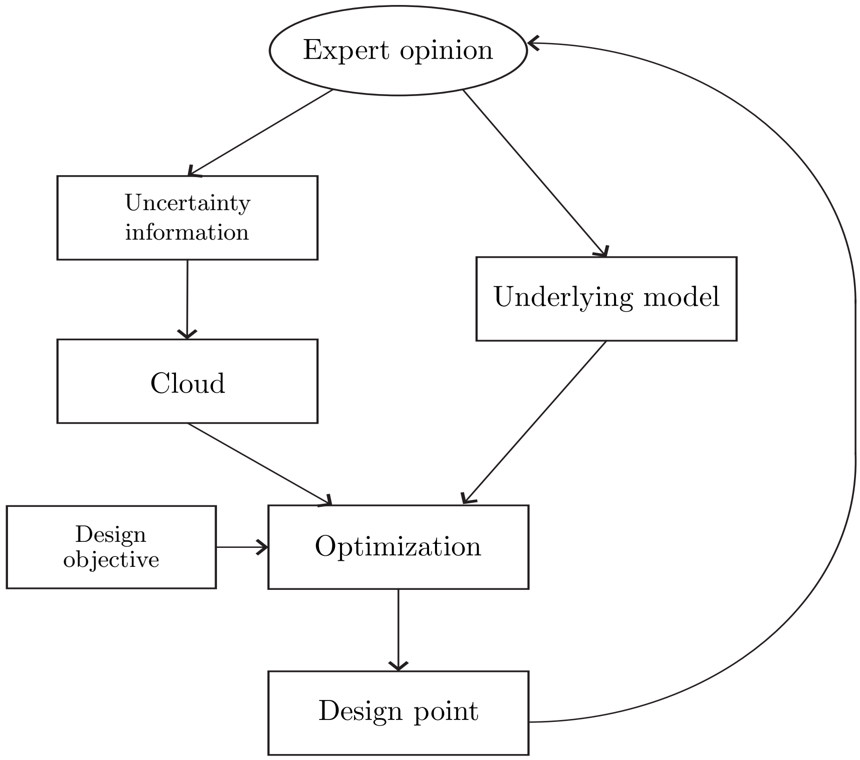

Figure 1. Basic concept.

Focussing on application examples from early phase spacecraft design, we will deal with a limited amount of uncertainty information, provided on the one hand as bounds or marginal probability distributions on the uncertain variables, without any formal correlation information. On the other hand, the engineers can adaptively improve the uncertainty model, even if their expert knowledge is only little formalized, by adding correlation constraints to exclude scenarios deemed irrelevant. The information can also be provided as real sample data, if available.

This paper is organized as follows. In Section 2 we present a more detailed study of uncertainty modeling with clouds. This is used to investigate robust design optimization, cf. Section 3. The techniques are applied to an example from spacecraft system design, described in Section 4. In Section 5 we discuss general and detailed aspects of our approach and conclude with a summary of our results.

2. Uncertainty modeling with clouds

The clouds formalism will serve as the central theoretical background for our uncertainty handling. Clouds will allow us an interpretation of uncertainties in terms of safety constraints. An important additional aspect of clouds is the ability to deal with high dimensional and non-formalized uncertainties.

This section starts with the definition of clouds in Section 2.1. The special case of potential clouds will be introduced as particularly interesting in Section 2.2, Section 2.3 will give an introduction about potential cloud generation.

2.1. Theoretical background

We start with the formal definition of clouds and introduce the notations. Let $\varepsilon \in \mathbb{M} \subseteq \mathbb{R}^n$ be an n-dimensional vector of uncertainties, we call ε an uncertain scenario. A *cloud* is a mapping $\chi(\varepsilon) = [\underline{\chi}(\varepsilon), \overline{\chi}(\varepsilon)]$, where $\chi(\varepsilon)$ is a nonempty, closed and bounded interval $\in [0,1]$ for all $\varepsilon \in \mathbb{M}$, and $]0,1[\subseteq \bigcup_{\varepsilon\in\mathbb{M}} \chi(\varepsilon) \subseteq [0,1]$. We call $\overline{\chi}(\varepsilon) - \underline{\chi}(\varepsilon)$ the width of the cloud χ. A cloud is called thin if it has width 0, and continuous if the lower level $\underline{\chi}$ and the upper level $\overline{\chi}$ are continuous functions of ε.

There exists a close relationship between thin continuous 1-dimensional clouds and cumulative distribution functions (CDFs) of real univariate random variables ε which is stated in Proposition 4.1 in (Neumaier, 2004): Let $F_\varepsilon(x) = \Pr(\varepsilon \leq x)$ be the CDF of ε, then $\chi(x) := F_\varepsilon(x)$ defines a thin cloud and $\Pr(\chi(\varepsilon) \leq y) = y, y \in \mathbb{M}$. The latter refers just to the fact that $F_\varepsilon(x)$ is uniformly distributed.

CDFs are well known from probability theory. Especially the 1-dimensional case is computationally unproblematic 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 $V : \mathbb{M} \to \mathbb{R}$.

2.2. 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. The idea is to construct a cloud from an interval-valued function χ of a user-defined potential function V, i.e., $\chi \circ V : \mathbb{M} \to [a,b]$, where $[a,b]$ is an interval in $[0,1]$.

Define the mapping

$$\chi(x) := [\underline{\alpha}(V(x)), \overline{\alpha}(V(x))], \tag{1}$$

where $\underline{\alpha}(y) := \Pr(V(\varepsilon) < y)$, $\overline{\alpha}(y) := \Pr(V(\varepsilon) \leq y)$, $\varepsilon \in \mathbb{M}$ a random variable. Then we get from Theorem 4.3 in (Neumaier, 2004) that we thus constructed a cloud χ that gives us an important interpretation in terms of confidence regions for ε.

Let $\alpha \in [0,1]$ be a given confidence level. The remarks to Theorem 4.3 in (Neumaier, 2004) tell us that if we choose $\underline{\alpha}(y)$ as a lower bound for $\Pr(V(\varepsilon) < y)$ and $\overline{\alpha}(y)$ as an upper bound for $\Pr(V(\varepsilon) \leq y)$, $\underline{\alpha}$, $\overline{\alpha}$ smooth and monotone, then χ as defined above is still a cloud. An appropriate bounding $\underline{\alpha}$, $\overline{\alpha}$ can be found, e.g., by Kolmogoroff-Smirnov (KS) statistics (Kolmogoroff, 1941). Then we define

$$\underline{C}_\alpha := \{\varepsilon | V(\varepsilon)) \leq \underline{V}_\alpha\}, \tag{2}$$

if a solution $\underline{V}_\alpha$ of $\overline{\alpha}(\underline{V}_\alpha) = \alpha$ exists and $\underline{C}_\alpha := \emptyset$ otherwise; analogously

$$\overline{C}_\alpha := \{\varepsilon | V(\varepsilon)) \leq \overline{V}_\alpha\}, \tag{3}$$

if a solution $\overline{V}_\alpha$ of $\underline{\alpha}(\overline{V}_\alpha) = \alpha$ exists and $\overline{C}_\alpha := \mathbb{M}$ otherwise. These are nested families of confidence regions parameterized by α: The region $\underline{C}_\alpha$ contains at most a fraction of α of all scenarios in $\mathbb{M}$, since $\Pr(\varepsilon \in \underline{C}_\alpha) \leq \Pr(\overline{\alpha}(V(\varepsilon)) \leq \alpha) \leq \Pr(F(V(\varepsilon)) \leq \alpha) = \alpha$; analogously $\overline{C}_\alpha$ contains at least a fraction of α of all scenarios in $\mathbb{M}$.

2.3. Potential cloud generation

Let's 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. We will investigate how to find these bounds. But first we consider the question how to choose the potential function. There are endless possibilities (see, e.g., Figure 2) to make the choice.

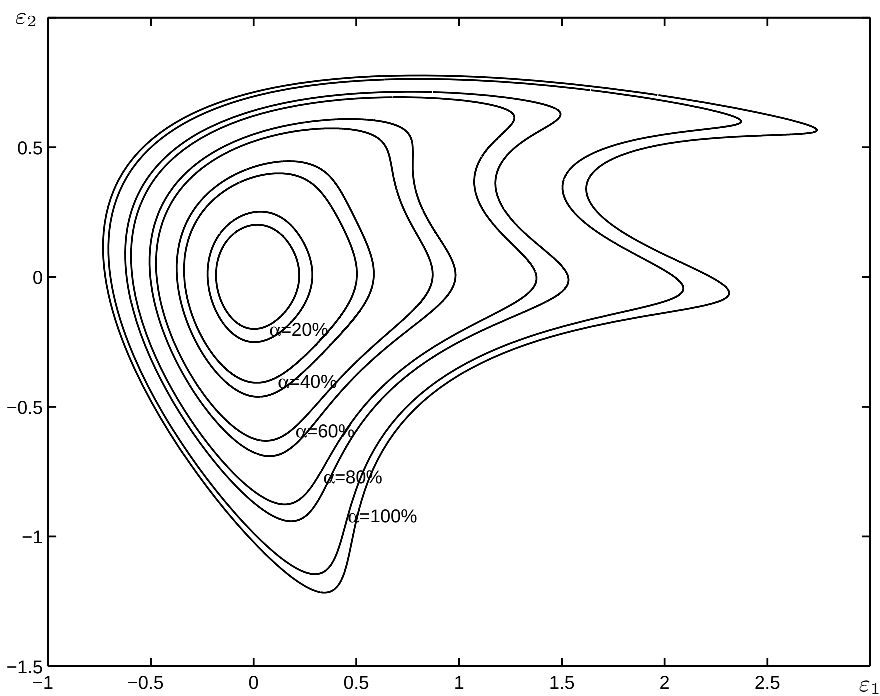

Figure 2. Nested confidence regions for the example of a 2-dimensional potential cloud, $\alpha = 0.2, 0.4, 0.6, 0.8, 1$.

Two special cases for choices of the potential function are

$$V(\varepsilon) := \max_k \frac{|\varepsilon^k - \mu^k|}{r^k}, \tag{4}$$

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

$$V(\varepsilon) := \|A\varepsilon - b\|_2^2, \tag{5}$$

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

A good choice of the potential should allow for a simple computational realization of the confidence regions, e.g., by linear constraints represented by $A\varepsilon \leq b$. This leads us to the investigation of polyhedron-shaped potentials, a generalization of box-shaped potentials. A polyhedron potential can be defined as:

$$V(\varepsilon) := \max_k \frac{(A\varepsilon)^k}{b^k}, \tag{6}$$

where $(A\varepsilon)^k, b^k$ are the k^{th} components of the vectors $(A\varepsilon)^k$ and b^k, respectively.

But how to achieve a polyhedron that reflects the given uncertainty information in the best way? As mentioned we assume the uncertainty information to consist of given samples, boxes or marginal distributions, and unformalized correlation constraints. After generation of a sample S as described later we define a box b_0 containing all sample points, and we define our potential $V_0(\varepsilon)$ box-shaped taking the value 1 on the margin of b_0.

Based on expert knowledge, a user-defined variation of V_0 can be performed by cutting off sample points deemed irrelevant for the worst-case. The exclusion of sample points is given by linear constraints $A\varepsilon \leq b$. Thus an expert can specify the uncertainty information in the form of linear correlation bounds adaptively resulting in a polyhedron shaped potential (6), even if the expert knowledge is only little formalized.

The adaptive exclusion of irrelevant scenarios, cf. Figure 3, can be realized in a graphical user interface (GUI). This procedure imitates iterative improvement in common real life MDO.

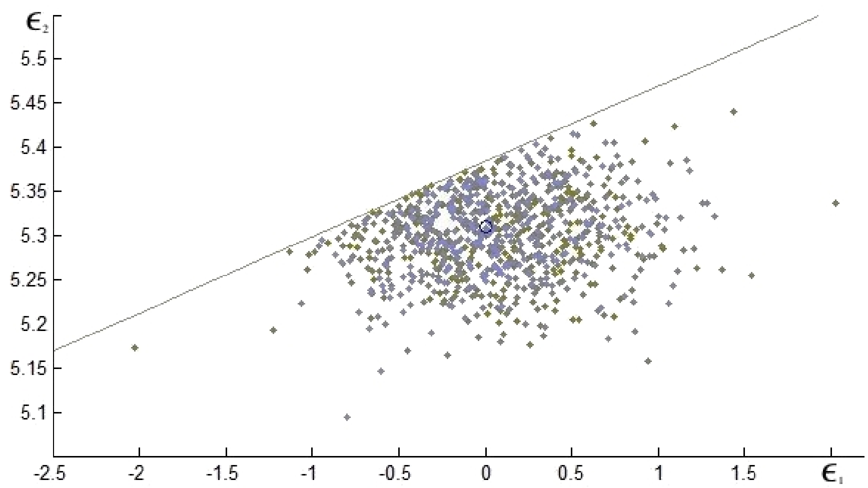

Figure 3. Exclusion of irrelevant scenarios by correlation bounds.

Now we turn to the investigation on how to find appropriate bounds on the CDF $F(V(\varepsilon))$. 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 KS statistics as suggested before. 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. 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, cf. (McKay et al., 1979), inspired method to generate the sample $S = \{x_1, \dots, x_{N_S}\}$ of N_S sample points. The sample points are chosen from a grid fulfilling the well-known Latin hypercube condition. If only boxes are given, then the grid is equidistant, if marginal distributions are given the grid is transformed with respect to them to ensure that each grid interval has the same marginal probability. Thus the generated sample represents the marginal distributions. However after a modification of S, e.g., by cutting off sample points as described, an assignment of weights to the sample is necessary to preserve the marginal CDFs.

In order to do so the weights $\omega_1, \dots, \omega_{N_S} \in [0, 1]$ are required to satisfy the following conditions: Let π_j be a sorting permutation of $\{1, \dots, N_S\}$, such that $x^j_{\pi_k(1)} \leq \dots \leq x^j_{\pi_k(N_S)}$. Let I be the index set of those entries of the uncertainty vector ε where a marginal CDF $F_i, i \in I \subseteq \{1, \dots, n\}$ is given. Then the weights should satisfy (7) $\forall i \in I, k = 1, \dots, N_S$

$$\sum_{j=1}^{k} \omega_{\pi_i(j)} \ \in [F_i(x^i_{\pi_i(k)}) - d, F_i(x^i_{\pi_i(k)}) + d], \ \sum_{k=1}^{N_S} \omega_k = 1. \tag{7}$$

The function

$$\widetilde{F}_i(\xi) \ := \textstyle\sum_{\{j \mid x^i_j \leq \xi\}} \omega_j \tag{8}$$

is a weighted marginal empirical distribution. For trivial weights, $\omega_1 = \dots = \omega_{N_S} = \frac{1}{N_S}$, $\widetilde{F}_i$ is a standard empirical distribution. The constraints (7) 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 $d = d_{\mathrm{KS}}$ with KS statistics:

$$d_{\mathrm{KS}} = \frac{\phi^{-1}(\alpha_{\mathrm{KS}})}{\sqrt{N_S} + 0.12 + \frac{0.11}{\sqrt{N_S}}}, \tag{9}$$

where ϕ is the Kolmogoroff function, α_{KS} the confidence in the KS theorem, cf. (Kolmogoroff, 1941), (Press et al., 1992).

Assume we have achieved weights satisfying (7), this yields a weighted empirical distribution

$$\widetilde{F}(\xi) \ := \textstyle\sum_{\{j \mid V(x_j) \leq \xi\}} \omega_j \tag{10}$$

approximating the CDF of $V(\varepsilon)$. If weights satisfying (7) can only be achieved with $d > d_{\mathrm{KS}}$, the relaxation d gives us an indicator for the quality of the approximation which will be useful to construct bounds on the CDF $F(V(\varepsilon))$.

After the approximation of $F(V(\varepsilon))$ with $\widetilde{F}$ we are just one step away from generating a potential cloud. Remember that we seek an appropriate bounding on $F(V(\varepsilon))$. We define $\overline{F} := \min(\widetilde{F} + D, 1)$

and $\underline{F} := \max(\widetilde{F} - D, 0)$, where D is computed with help of the KS approach (9), and fit these two step functions to smooth, monotone lower bounds $\underline{\alpha}(V(\varepsilon))$ and upper bounds $\overline{\alpha}(V(\varepsilon))$. If the 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 bounding of the CDF $F(V(\varepsilon))$ and according to the remarks to Theorem 4.3. in (Neumaier, 2004) mentioned we have generated a potential cloud that fulfills the conditions that define a cloud via the mapping $\chi : \varepsilon \to [\underline{\alpha}(V(\varepsilon)), \overline{\alpha}(V(\varepsilon))]$.

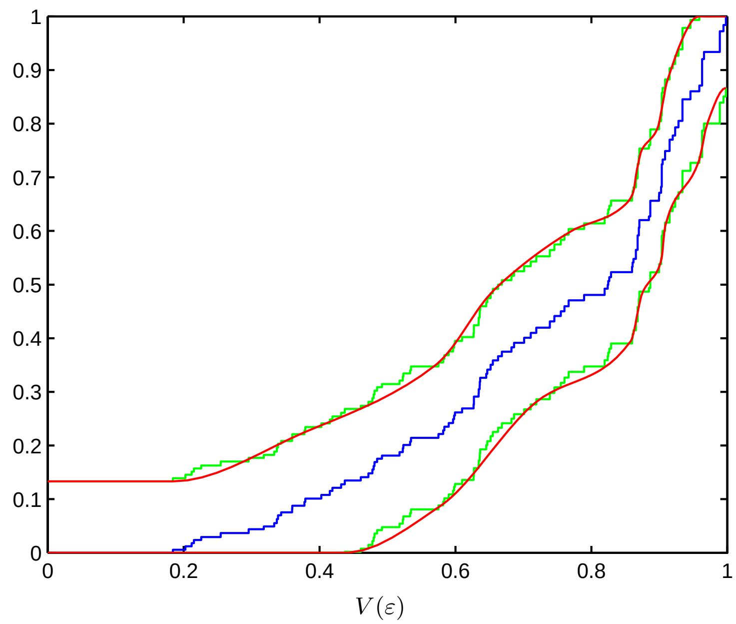

Figure 4. The smooth lower bounds $\underline{\alpha}(V(\varepsilon))$ and upper bounds $\overline{\alpha}(V(\varepsilon))$ for a potential cloud.

The cloud represents the given uncertainty information and now enables us to interpret the potential level maps $\{\varepsilon | V(\varepsilon) = V_\alpha\} = \underline{C}_\alpha$ as *confidence regions* for our uncertain vector ε. They are the worst-case relevant regions.

Hence the 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, CDF approximations with empirical distributions, KS statistics to estimate bounds, sample generation methods, and weighting techniques.

3. Robust design optimization

A classic approach to design optimization, without taking uncertainties into account, leads to decision support for engineers, but to a design which completely lacks robustness. We want to safeguard the design against uncertain errors. That will involve the methods for uncertainty modeling we introduced in the last section.

First we give a formal statement of the optimization problem in Section 3.1. Afterwards we point out the difficulties related in Section 3.2 and finally present a solution approach in Section 3.3.

3.1. Problem formulation

Provided an underlying model of a given structure like a spacecraft component, with several inputs and outputs, we denote as x the vector containing all output variables, and as 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 $\theta^1, \dots, \theta^{n_o}$. 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, $I_d \cup I_c = \{1, \dots, n_o\}$, $I_d \cap I_c = \emptyset$. 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 $1, \dots, N_i$ 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 $(v_1^i, \dots, v_{n_i}^i) \in \mathbb{R}^{n_i}$. Usually this set is provided in a $N_i \times n_i$ table (see, e.g., Table II, $N_i = 30$, $n_i = 3$). In the continuous case, $i \in I_c$, the choice variable θ^i can be regarded as a design variable in a given interval $[\underline{\theta^i}, \overline{\theta^i}]$. 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. Let $Z(\theta)$ be a mapping assigning an input vector z to the design choice θ. We call Z a table mapping as the nontrivial parts of Z consist of tables.

Both design and global input variables contained in z can be uncertain, ε denotes the related vector of uncertainties. We assume that the optimization problem can be formulated as a mixed-integer, bi-level problem of the following form:

$$
\begin{array}{lll}
\min\limits_{\theta} \ \max\limits_{x,z,\varepsilon} & g(x) & \text{(objective functions)} \\
\text{s.t.} & z = Z(\theta) + \varepsilon & \text{(table constraints)} \\
 & G(x,z) = 0 & \text{(functional constraints)} \\
 & \theta \in T & \text{(selection constraints)} \\
 & V(\varepsilon) \leq \underline{V}_\alpha & \text{(cloud constraint)}
\end{array}
\tag{11}
$$

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 whose value is the nominal entry from $Z(\theta)$ plus its error ε with uncertainty specified by the cloud. 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., $\theta^i \in \{1, \dots, N_i\}$ if $i \in I_d$ and $\theta^i \in [\underline{\theta^i}, \overline{\theta^i}]$ if $i \in I_c$. The cloud constraint involves the potential function V as described in the Section 2 and models the worst-case relevant region $\{\varepsilon | V(\varepsilon) \leq \underline{V}_\alpha\} = \underline{C}_\alpha$.

3.2. 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 program (MINLP). MINLP is still a recent research direction which has not yet matured. 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. Additionally we face a bi-level structure imposed by the uncertainties, which is already a nontrivial complication in the traditional situation where all variables are continuous. 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 (11).

In view of these difficulties we are limited to the use of heuristic methods, i.e., we treat the functional constraints of the underlying model as a black-box function $x = G_{\mathrm{bb}}(z)$ and make use of specific strategies to sample from the set of allowed inputs $z = Z(\theta)$, $\theta \in T$.

3.3. Solution approach

We will first reformulate the problem incorporating the objective function and functional constraints for the underlying model in the black-box function $G_{\mathrm{bb}}(z)$.

$$\begin{aligned} \min_{\theta} \ \max_{z,\varepsilon} \quad & G_{\mathrm{bb}}(z) \\ \text{s.t.} \quad & z = Z(\theta) + \varepsilon \\ & \theta \in T \\ & V(\varepsilon) \leq \underline{V}_\alpha \end{aligned} \tag{12}$$

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

$$\begin{aligned} \max_{z,\varepsilon} \quad & G_{\mathrm{bb}}(z) \\ \text{s.t.} \quad & z = Z(\theta) + \varepsilon \\ & V(\varepsilon) \leq \underline{V}_\alpha \end{aligned} \tag{13}$$

Because of the polyhedral structure of our clouds, the cloud constraint $V(\varepsilon) \leq \underline{V}_\alpha$ can be written as a collection of linear inequalities parameterized by the confidence level α. We approximate G_{bb} in a small box containing the region $\{\varepsilon | V(\varepsilon) \leq \underline{V}_\alpha\}$ linearly. Thus problem (13) becomes an LP solved by an LP solver, cf. (Grant and Boyd, 2007). The maximizer $\widehat{\varepsilon}$, $\widehat{z} = Z(\theta) + \widehat{\varepsilon}$ for the fixed design choice θ corresponds to the worst-case objective function value $\widehat{G}_{\mathrm{bb}}(\theta) := G_{\mathrm{bb}}(Z(\theta) + \widehat{\varepsilon})$. The function $\theta \to \widehat{G}_{\mathrm{bb}}(\theta)$ implicated by the solution of problem (13) is now used to get rid of the bi-level structure in problem (12):

$$\begin{array}{ll} \min\limits_{\theta} & \widehat{G}_{\mathrm{bb}}(\theta) \\ \text{s.t.} & \theta \in T \end{array} \tag{14}$$

The method we develop to solve this 1-level problem, and to seek the robust, optimal design, 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. Remember θ is the vector of design choice variables $\theta^1, \dots, \theta^{n_o}$. We look for a separable underestimator $q(\theta)$ for the objective function of the form:

$$q(\theta) := \sum_{i=1}^{n_o} q_i(\theta^i). \tag{15}$$

Let $\theta \in T$, $z = Z(\theta)$. Assume the black-box G_{bb} has been evaluated N_o times resulting in the function evaluations $G_{\mathrm{bb}_1}, \dots, G_{\mathrm{bb}_{N_o}}$ for the design choices $\theta_1, \dots, \theta_{N_0}$. Let $l \in \{1, \dots, N_o\}$. For a discrete choice θ_l^i, $i \in I_d$, we define $q_i(\theta_l^i) := q_{i,\theta_l^i}$, $\theta_l^i \in \{1, \dots, N_i\}$, simply as a constant. For a continuous choice θ_l^i, $i \in I_c$, we define $q_i(\theta_l^i) := \theta_l^i \cdot q_{i1} + \theta_l^i \cdot q_{i2}^2$ by a quadratic expression with the two constants q_{i1} and q_{i2}. If $I_d = \emptyset$ we add an integer choice θ^i with $N_i = 1$ artificially to represent the constant part which is missing 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 a linear optimization program (LP) satisfying the constraints

$$\sum_{i=1}^{n_o} q_i(\theta_l^i) \ \leq G_{\mathrm{bb}_l} \quad l = 1, \dots, N_o \tag{16}$$

and ensuring that many constraints in (16) will be active. The underestimator $q(\theta)$ is separable and can be easily minimized.

Apart from the method of separable underestimation we also make use of further strategies to find a solution of the optimization problem (14). The first one fits a quadratic model for the G_{bb} which is minimized afterwards, cf. (Huyer and Neumaier, 2006). Integers are treated as continuous variables and rounded to a grid with step width 1. Another method is based on evolution strategy with covariance matrix adaptation, cf. (Hansen and Ostermeier, 2001). It is a stochastic method to sample the search space. Integers are also treated as continuous variables rounded to the next integer value.

Finally the minimizers that result from all methods used are starting points for a limited global search that consists of an integer line search for the discrete choice variables and multilevel coordinate search (Huyer and Neumaier, 1999) for the continuous choice variables. Thus we hope to find the global optimal solution, but as we are using heuristics there is no guarantee.

Remark. For the implementation of our methods we formulated them as MATLAB code. The following is a summary of all external routines we use in our methods: we make use of the Statistics Toolbox of MATLAB to evaluate probability distributions; we use CVX (Grant and Boyd, 2007) to solve linear programs; SNOBFIT (Huyer and Neumaier, 2006) and MCS (Huyer and Neumaier,

1999) as external optimization routines; NLEQ (Deuflhard, 2004), (Nowak and Weimann, 1990) to solve systems of nonlinear equations.

4. Application example

Here we apply our methods for robust and autonomous design to a case study of early phase spacecraft engineering, i.e., the Attitude Determination and Control Subystem (ADCS) for the NASA's Mars Exploration Rover (MER) mission cf. (MER, 2003), (Erickson, 2004) whose scientific goal is to investigate the history of water on 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 II) 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. 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 the notations introduced, the choice variable θ, i.e., the type of thruster, can be selected as an integer between 1 and 30.

Uncertainty specifications, variable structure, the MER mission maneuver sequence, and system model equations to compute the total mass m_{tot} are taken from (Thunnissen, 2005). The uncertainty specification for the model variables are reported in Table III of Appendix C. 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 A. Moreover, a survey on the system model equations and the MER mission sequence can be found in the Appendices of (Fuchs et al., 2007).

4.1. Results

The cloud constraints for the optimization are generated for a confidence level of $\alpha = 95\%$ and a generated sample size $N_S = 1000$. The results for optimization are divided into four different configurations of uncertainty handling and specifications:

a. The uncertainties are as specified in Table III. Here we treat them in a classical engineering way, assigning 3 σ boxes 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 III. With our methods we find the optimal design choice $\theta = 9$ as in Configuration 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 configuration we do not take any uncertainties into account, generally assuming the nominal case for all uncertain input variables. 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 Configuration b.

d. The uncertainties are obtained by taking the values from Table III 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 I, showing the optimal design choice for each configuration and the corresponding value of the objective function m_{tot} for the nominal case and for the worst case, respectively.

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

Configuration	**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

The results show a number of important facts related to spacecraft design. The comparison between the configurations b and d suggests that in a preliminary stage of the spacecraft systems modeling 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 new information becomes continuously available. Our method captures this important dynamics and processes it in rigorous mathematical terms.

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

5. Discussion & Conclusions

The importance of robustness in design optimization has been the starting point and main motivation of our research work, and our results on a case study confirm that the optimal spacecraft design is strongly sensitive to uncertainties. At the present stage we can clearly state that neglecting uncertainties results in a design that completely lacks robustness and a simplified uncertainty model (like a 3 σ approach) may yield critical underestimations of worst-case scenarios.

When trying to collect the uncertainty information, it turned out to be very difficult to get useful information directly from expert engineers. To collect the information, an interactive dialogue between the experts and the computer can be realized by a GUI where the engineers can specify uncertainties, provide sample data, cut off worst-case irrelevant scenarios, and adjust the quality of the uncertainty model. We expect that this kind of interaction is an inevitable next step in design processes, especially spacecraft design. We continue the discussion with more detailed considerations on the study.

- In the theory of clouds, cf. Section 2 and (Neumaier, 2004), there is a distinction between the confidence regions of α-relevant scenarios $\underline{C}_\alpha$, α-reasonable scenarios $\overline{C}_\alpha$ and borderline cases (which is the set difference of the α-reasonable and the α-relevant regions). In robust design the possibly uncertain scenarios are required to satisfy safety constraints. With respect to our terminology the regions above have the following interpretation: if at least one of the α-relevant scenarios fails to satisfy the safety constraints, the design is unsafe; if all of the α-reasonable scenarios satisfy the safety constraints, the design is safe. Between these two cases there is the borderline region where no precise statement can be made without additional uncertainty information. The volume of the borderline region is increasing if the width of the cloud increases and vice versa. So widening the cloud enlarges the borderline region, corresponding to a lack of uncertainty information. This fact is reflected in our approach as both a smaller sample size and an increased dimension of the uncertainty result in a wider cloud.
- The width of the cloud is defined as the difference between the mappings $\underline{\alpha}$ and $\overline{\alpha}$ (cf. Section 2). We constructed the mappings to fulfill the conditions that define a cloud with an algorithm which is non-rigorous, but has a high, adjustable reliability. Thus the user of the algorithm is able to control the desired level of reliability.
- As mentioned before the reliability of our worst-case analysis with clouds is determined by user-defined parameters, i.e., the size of the generated sample S and confidence levels for sample generation, CDF bounding and approximation. Concerning the sample size: if we increase the size of S we artificially refine the uncertainty model and get more reliability of the worst-case analysis. A larger sample is computationally more expensive, in particular the weight computation, so the reliability is also a trade-off with performance.
- The choice of the potential function is arbitrary. Different shapes of the cloud (i.e., shapes of the potential) can make the worst-case analysis more pessimistic or optimistic. We point out that a poor choice of the potential makes the worst-case analysis more pessimistic, but will still result in a valid robust design. We allow a variation of the potential by switching from a box-shaped to a polyhedron-shaped potential to enable the experts to improve the uncertainty model iteratively.

- A good weight computation (cf. Section 2.3) is the key to a good uncertainty representation with clouds. In higher dimensions the weight computation is very expensive. To overcome this problem and to allow the adjustment of the computation time, the relaxation radius d must be increased carefully. In our algorithm we respect the relaxation property, widening the cloud by the amount of relaxation after evaluating the quality of the weights as described in Section 2.3.
- As mentioned before, we are limited to the use of heuristic methods since the design problem (11) is highly complex and not suitable for standard optimization methods. In our problem formulation we seek the design with the optimal worst-case scenario. 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.
- The number 34 of uncertain variables in our case study is large enough to make our problem representative for uncertainty handling in real-life applications.
- Though global optimality for the solution in our application example is very likely, as the choice variable is 1-dimensional and discrete, in general the heuristical methods cannot guarantee global optimality of the problem solution.
- The approach with separable underestimation introduced in this chapter takes advantage of inherent characteristics of spacecraft design problems, i.e., the discrete nature of many of the variables involved, supporting, at the same time, continuous choice variables. Details on our heuristic methods for design optimization introduced in Section 3 will be published elsewhere.

5.1. Conclusions and Future Work

In this chapter we presented a new approach to autonomous robust design optimization. Starting from the background of the cloud theory we developed methodologies to process the uncertainty information from expert knowledge towards a reliable worst-case analysis and an optimal and robust design. Our approach is applicable to real-life problems such as, e.g., early phase spacecraft system design. In the example of the community of spacecraft engineers, at present, in most instances of the design process, reliability is only assessed qualitatively by the experts. We present a step forward towards quantitative statements about the design reliability.

The adaptive nature is one of the key features of our uncertainty model as it imitates real-life design strategies. The iteration steps significantly improve the uncertainty information and we are able to process the new information to an improved uncertainty model.

The presented approach is generally applicable to problems of robust design optimization, not only spacecraft design. In particular problems with discrete design choices can be tackled. The advantages of achieving the optimal design autonomously is undeniable. Though we already applied the new methods to different design problems, cf. (Neumaier et al., 2007), one future goal is to apply them to more problem classes in order to learn from new challenges.

With our approach we can process the available uncertainty information to perform a reliable worst-case analysis linked to an adjustable confidence level. An additional value of the uncertainty model is the fact that one can capture various forms of uncertainty information, even those less formalized. There is no loss of valuable information, and the methods are capable of handling the uncertainties reliably, even if the amount of information is very limited.

Summing up, the presented methods offer an exciting novel approach to face the highly complex problem of autonomous robust design optimization, an approach which is easily understandable, reliable and computationally realizable.

Appendix

A. Model Variable Structure

Remark. Do not confuse the notations in these appendices with our notation of the main sections.

The 47 variables involved in the model fall into the following four categories:

– **5 constant parameters**.
  Input variables for the model with fixed values and no uncertainty.

Constant parameter	Description	Value
c_0	speed of light in a vacuum	299792458 m/s
d	average distance from the spacecraft to the sun in AU	1.26 AU
g_0	gravity constant	9.8 m/s^2
t	total mission time	216 days
θ_i	sunlight angle of incidence	0°

– **33 Uncertain input variables**.
  The uncertainties are specified by probability distributions for each of these variables (cf. Appendix C).

Variable	Description
A_{max}	maximal cross-sectional area
J_{xx}, J_{zz}	moments of inertia
R	engine moment arm
δ_1, δ_2	engine misalignment angle
g_s	solar constant at 1 AU
κ	distance from the center of pressure to the center of mass
ω_{spin_i}	spin rates, $i = 0...3$, given in rpm
ψ_{slew_i}	slew angles, $i = 1...19$, given in $^\circ$
q	spacecraft surface reflectivity
$uncfuel$	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).

Variable	Description
F	thrust
I_{sp}	specific impulse
m_{thrust}	mass of a thruster

– **6 Result variables.**
Result variables containing the objective for the optimization m_{tot}.

Variable	Description
m_{fp}	fuel mass needed for fault protection maneuvers
m_{fuel}	total fuel mass needed for all maneuvers
m_{slew}	fuel mass needed for slew maneuvers
m_{slew_s}	fuel mass needed for slew maneuvers fighting solar torque
m_{spin}	fuel mass needed for spin maneuvers
m_{tot}	total mass of the subsystem

B. Thruster specification

Table II 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: the table represents 30 discrete choices in $\mathbb{R}^3$. The 3 design variables are the 3 components of these points in $\mathbb{R}^3$. The choice variable θ is 1-dimensional and has an integer value between 1 and 30. The various sources for the data contained in Table II are (EADS, 2007), (Thunnissen, 2005), (Purdue School of Aeronautics and Astronautics, 1998), (Zonca, 2004), (Personal communication, 2007).

C. Uncertainty specification

All uncertainty specifications taken from (Thunnissen, 2005) are reported in Table III. The notation used for the probability distributions is:

Notation	Distribution
$U(a, b)$	uniform distribution in (a, b)
$N(\mu, \sigma)$	normal distribution with mean μ and variance σ^2
$\Gamma(\alpha, \beta)$	gamma distribution with mean $\alpha\beta$ and variance $\alpha\beta^2$
$L(\mu, \sigma)$	lognormal distribution, distribution parameters μ and σ (mean and standard deviation of the associated normal distribution)

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 II) and standard deviation $\frac{7}{300}F_{table}$.

Table II. Thruster specifications and the linked choice variable θ.

θ	**Thruster**	F/N	I_{sp}/s	m_{thrust}/kg
1	Aerojet MR-111C	0.27	210	0.2
2	EADS CHT 0.5	0.5	227.3	0.195
3	MBB Erno CHT 0.5	0.75	227	0.19
4	TRW MRE 0.1	0.8	216	0.5
5	Kaiser-Marquardt KMHS Model 10	1	226	0.33
6	EADS CHT 1	1.1	223	0.29
7	MBB Erno CHT 2.0	2	227	0.2
8	EADS CHT 2	2	227	0.2
9	EADS S4	4	284.9	0.29
10	Kaiser-Marquardt KMHS Model 17	4.5	230	0.38
11	MBB Erno CHT 5.0	6	228	0.22
12	EADS CHT 5	6	228	0.22
13	Kaiser-Marquardt R-53	10	295	0.41
14	MBB Erno CHT 10.0	10	230	0.24
15	EADS CHT 10	10	230	0.24
16	EADS S10 - 01	10	286	0.35
17	EADS S10 - 02	10	291.5	0.31
18	Aerojet MR-106E	12	220.9	0.476
19	SnM 15N	15	234	0.335
20	TRW MRE 4	18	217	0.5
21	Kaiser-Marquardt R-6D	22	295	0.45
22	Kaiser-Marquardt KMHS Model 16	22	235	0.52
23	EADS S22 - 02	22	290	0.65
24	ARC MONARC-22	22	235	0.476
25	ARC Leros 20	22	293	0.567
26	ARC Leros 20H	22	300	0.4082
27	ARC Leros 20R	22	307	0.567
28	MBB Erno CHT 20.0	24	234	0.36
29	EADS CHT 20	24.6	230	0.395
30	Daimler-Benz CHT 400	400	228.6	0.325

References

J. A. Aguilar, A. B. Dawdy, and G. W. Law. The aerospace corporations concept design center. In *8th Annual International Symposium of the International Council on Systems Engineering*, 1998.

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*, 1997.

Table III. ADCS uncertainty specifications.

Variable	Probability Distribution	Variable	Probability Distribution
A_{max}	$N(5.31, 0.053)$	ψ_{slew5}	$N(2.76, 0.2)$
J_{xx}	$U(300, 450)$	ψ_{slew6}	$N(8.51, 0.4)$
J_{zz}	$U(450, 600)$	ψ_{slew7}	$N(9.88, 0.5)$
R	$N(1.3, 0.0013)$	ψ_{slew8}	$N(5.64, 0.2)$
δ_1	$N(0, 0.5)$	ψ_{slew9}	$N(5.04, 0.2)$
δ_2	$N(0, 0.5)$	ψ_{slew10}	$N(5.75, 0.2)$
g_s	$N(1400, 14)$	ψ_{slew11}	$N(4.47, 0.1)$
κ	$U(0.6, 0.7)$	ψ_{slew12}	$N(5.53, 0.1)$
ω_{spin0}	$N(12, 1.33)$	ψ_{slew13}	$N(5.85, 0.1)$
ω_{spin1}	$N(2, 0.0667)$	ψ_{slew14}	$\Gamma(1.5, 10.5)$
ω_{spin2}	$\Gamma(11, 0.25)$	ψ_{slew15}	$\Gamma(1.5, 10.5)$
ω_{spin3}	$L(2, 0.0667)$	ψ_{slew16}	$\Gamma(1.5, 10.5)$
ω_{spin4}	$N(48, 5)$	ψ_{slew17}	$\Gamma(1.5, 10.5)$
ω_{spin5}	$N(2, 0.0667)$	ψ_{slew18}	$\Gamma(1.5, 10.5)$
ψ_{slew1}	$N(5, 0.5)$	ψ_{slew19}	$\Gamma(1.5, 10.5)$
ψ_{slew2}	$N(50.45, 5)$	q	$N(0.6, 0.06)$
ψ_{slew3}	$N(5.13, 0.5)$	$uncfuel$	$N(0, 0.05)$
ψ_{slew4}	$N(6.35, 0.6)$	F	$N(F_{table}, 7/300F_{table})$

V. Amata, G. Fasano, L. Arcaro, F. Della Croce, M. Norese, S. Palamara, R. Tadei, and F. Fragnelli. Multidisciplinary optimisation in mission analysis and design process. GSP programme ref: GSP 03/N16 contract number: 17828/03/NL/MV, European Space Agency, 2004.

M. Bandecchi, S. Melton, and F. Ongaro. *Concurrent engineering applied to space mission assessment and design*. ESA Bulletin, 1999.

V. Belton and T. J. Stewart. *Multiple criteria decision analysis: an integrated approach*. Kluwer Academic Publishers, 2002.

P. Deuflhard. *Newton Methods for Nonlinear Problems. - Affine Invariance and Adaptive Algorithms.*, volume 35 of *Series in Computational Mathematics*. Springer, 2004.

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

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

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

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.

S. Ferson. What monte carlo methods cannot do. *Human and Ecological Risk Assessment*, 2:990–1007, 1996.

S. Ferson, L. Ginzburg, and R. Akcakaya. Whereof one cannot speak: When input distributions are unknown. Risk Analysis, in press, 1996.
`http://www.ramas.com/whereof.pdf`.

M. Fuchs, D. Girimonte, D. Izzo, and A. Neumaier. Robust and autonomous space system design. submitted, 2007. http://www.mat.univie.ac.at/~mfuchs/files/raassd07.pdf.

M. C. Grant and S. P. Boyd. Cvx: A system for disciplined convex programming. 2007. http://www.stanford.edu/~boyd/cvx/cvx_usrguide.pdf http://www.stanford.edu/~boyd/cvx/.

N. Hansen and A. Ostermeier. Completely derandomized self-adaptation in evolution strategies. *Evolutionary Computation*, 9(2):159–195, 2001. http://www.bionik.tu-berlin.de/user/niko/cmaartic.pdf.

W. Huyer and A. Neumaier. Snobfit - stable noisy optimization by branch and fit. submitted preprint, 2006. http://www.mat.univie.ac.at/~neum/ms/snobfit.pdf http://www.mat.univie.ac.at/~neum/software/snobfit/.

W. Huyer and A. Neumaier. Global optimization by multilevel coordinate search. *J. Global Optimization 14*, pages 331–355, 1999. http://www.mat.univie.ac.at/~neum/software/mcs/.

G. Karpati, J. Martin, M. Steiner, and K. Reinhardt. The integrated mission design center (IMDC) at NASA goddard space flight center. In *IEEE Aerospace Conference*, volume 8, pages 3657–3667, 2003.

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.

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

V. Kreinovich. *Random Sets: Theory and Applications*, chapter Random sets unify, explain, and aid known uncertainty methods in expert systems, pages 321–345. Springer-Verlag, 1997.

W. J. Larson and J. R. Wertz. *Space Mission Analysis and Design.* Microcosm Press, third edition, 1999.

M. Lavagna and A. E. Finzi. A multi-attribute decision-making approach towards space system design automation through a fuzzy logic-based analytic hierarchical process. In *Proceedings of the 15th International Conference on Industrial and Engineering. Applications of Artificial Intelligence and Expert Systems*, 2002.

K. Lewis and F. Mistree. Modeling interactions in multidisciplinary design: A game theoretic approach. *AIAA Journal*, 35(8):1387–1392, 1997.

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

M. McKay, W. Conover, and R. 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.

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

A. Neumaier. On the structure of clouds. Manuscript, 2003. http://www.mat.univie.ac.at/~neum/ms/struc.pdf.

A. Neumaier. Clouds, fuzzy sets and probability intervals. *Reliable Computing 10*, pages 249–272, 2004. http://www.mat.univie.ac.at/~neum/ms/cloud.pdf.

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.

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. http://www.zib.de/Numerik/numsoft/CodeLib/codes/nleq1_m/nleq1.m.

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.

Personal communication with ESA engineers, 2007.

W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery. *Numerical recipes in C.* Cambridge University Press, second edition, 1992.

Purdue School of Aeronautics and Astronautics. Satellite Propulsion Web Page, 1998. http://cobweb.ecn.purdue.edu/~propulsi/propulsion/rockets/satellites.html.

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

B. Roy. *Multicriteria methodology for decision aiding*. Kluwer Academic Publishers, 1996.

G. Shafer. *A Mathematical Theory of Evidence*. Princeton, NJ: Princeton University Press, 1976.

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

R. C. Williamson. *Probabilistic Arithmetic*. PhD thesis, University of Queensland, 1989.

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, 2004.