Uncertainty modeling for robust verifiable design

Arnold Neumaier

University of Vienna

Vienna, Austria

Safety

Safety studies in structural engineering are supposed to guard against failure in all reasonable situations encountered during the lifetime of a structure.

What is 'reasonable' is inferred on the basis of knowledge about the past performance of structures and their building blocks.

The hope is that the future will be sufficiently like the past, so that the unknown future of a (planned or existing) structure can be assessed based on this past information.

Traditionally, the past information is summarized in form of either probability distributions (stochastic model), worst case bounds (interval model), or parameterized families of bounds (fuzzy set model).

Design

A general design problem involves the following variables:

- θ **(wanted, controllable, low-D) design vector**
- x **(uncontrollable, high-D) state vector**
- y **response vector (determined by** θ **and** x**)**
- c **cost vector**
- ε **failure probability**

x **contains variables for uncertain material parameters (length, cross section, elasticity module, etc.), uncertain model parameters, etc.**

Model of the design context

- $F_d(c,\theta) \in \mathbf{F}_d$ **design restrictions**
- $x \in X$ **uncertain state information**
- $E(\theta,x,y)=0$ **model equations** $(\dim E = \dim y)$
- $F_s(c,y) \in \mathbf{F}_s$ **response requirements**
- $q(c,\varepsilon) \geq q_{\text{target}}$ **quality constraint**

Notation: $F \in \mathbf{F} = [\underline{F}, \overline{F}] \quad \Leftrightarrow \quad \underline{F}_i \leq F_i \leq \overline{F}_i$ **for all** i

(includes equality constraints and one-sided inequalities)

Most real life design problems fall into this general framework. Cost-dependent constraints allow soft formulations!

Failure probability

The failure probability

$$\varepsilon := 1 - \Pr(F_s(c, y) \in \mathbf{F}_s \mid E(\theta, x, y) = 0)$$

is determined by the model only if the <u>complete</u> joint distribution of all state vector variables x_i is available.

This is <u>never</u> the case in real situations once x contains more than a few variables only.

People often substitute default independence or normality assumptions where in fact information is completely missing.

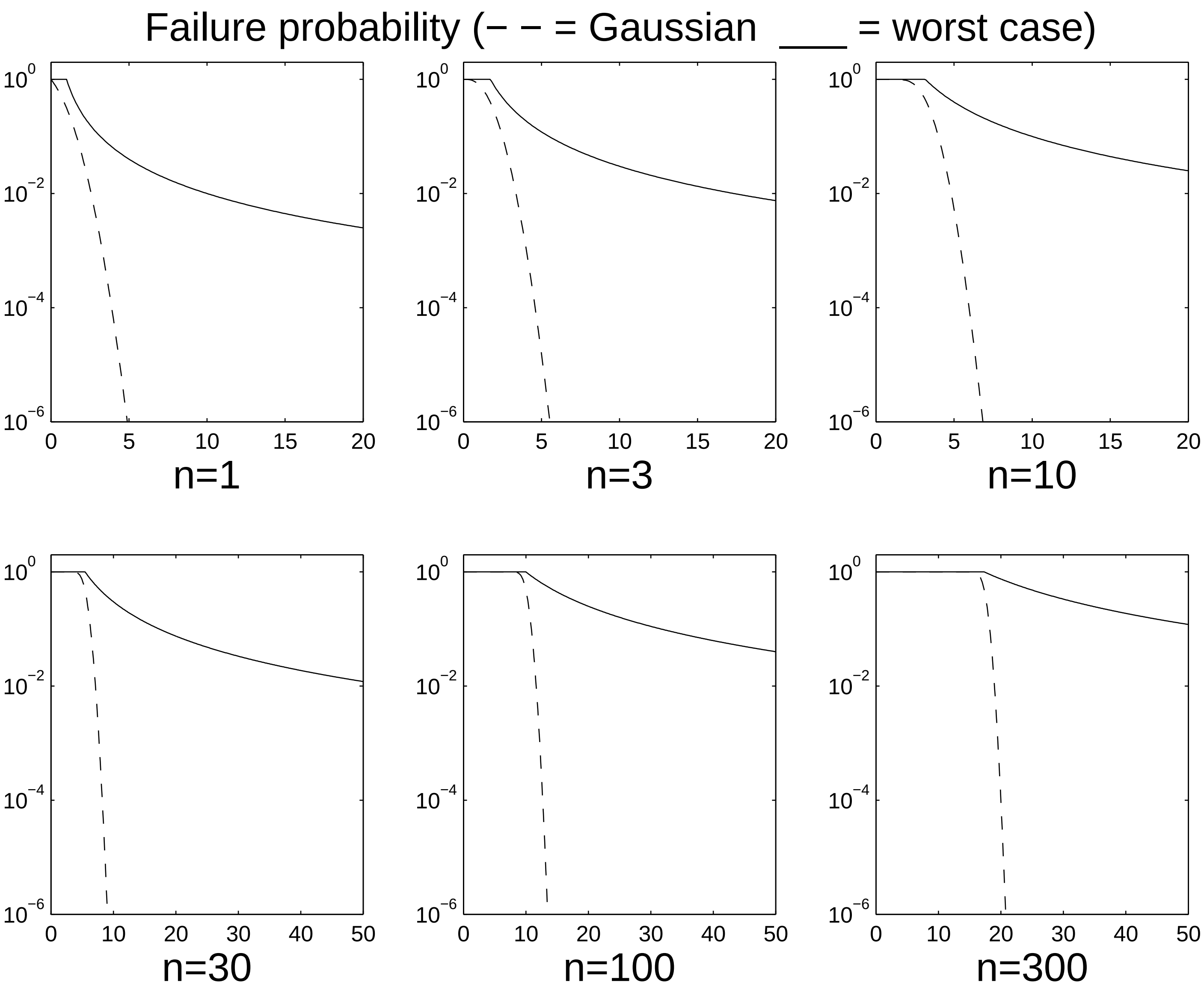
In particular, tails of the joint distribution cannot be estimated reliably without an excessive amount of data.

This may lead to drastically wrong assessments, since probabilistic design problems usually depend sensitively on these tails.

Theorem. *The best possible bound for the probability that an n-dimensional random vector x with uncorrelated coefficients of mean zero and variance 1 has length at least r is given by the generalized Chebyshev inequality*

$$\Pr(\|x\|_2 \geq r) \leq \min(1, n/r^2).$$

Although common practice, representing such a vector by a Gaussian distribution, while in fact the distribution is unknown, gives far too optimistic results.


Failure probability (− − = Gaussian ___ = worst case)
10^0
10^−2
10^−4
10^−6
0 5 10 15 20
n=1
n=3
n=10
0 10 20 30 40 50
n=30
n=100
n=300

Even if the complete distribution of x were known (or simply assumed), computing the failure probability ε accurately is very difficult, and nearly impossible in high dimensions.

Traditional approximations such as FORM and SORM often work well, but may fail without warning.

Quotes from the Bible

(Contemporary English Version)

We make our own plans, but the LORD decides where we will go. (Proverbs 16:9)

We make our own decisions, but the LORD alone determines what happens. (Proverbs 16:33)

**We may throw the dice, but the LORD determines how they fall.
(Proverbs 16:33, New Living Translation)**

We may make a lot of plans, but the LORD will do what he has decided. (Proverbs 19:21)

How can we know what will happen to us when the LORD alone decides? (Proverbs 20:24)

Probabilistic design goal

Maximize the design quality

$$\begin{array}{ll} \max & q(c,\varepsilon) \\ \text{s.t.} & F_d(c,\theta) \in \mathbf{F}_d \\ & \Pr(F_s(c,y) \in \mathbf{F}_s \mid E(\theta,x,y)=0) = 1-\varepsilon \end{array}$$

... and check whether the requested quality $q \geq q_0$ was obtained.

This is a stochastic program with an expensive, low accuracy (hence noisy) objective function, often multimodal!

$\Rightarrow$ SNOBFIT

www.mat.univie.ac.at/∼neum/software/snobfit/

Robust, verifiable design

replaces the inaccessible probabilistic model input by deterministic assumptions that can be justified reasonably well by a limited amount of data.

In particular, it avoids using information which in fact is not reliably available.

Within such a deterministic framework, a <u>rigorous worst case analysis</u> becomes possible.

The results are (in principle) verifiable.

In the verifiable design methodology, probabilistic methods are replaced by techniques of global optimization.

This provides guarantees for the model behavior that are as reliable as the information in the model and the description of the uncertain state.

Currently the best global solvers are BARON and OQNLP:
http://archimedes.scs.uiuc.edu/baron/baron.html
http://www.opttek.com/products/gams.html

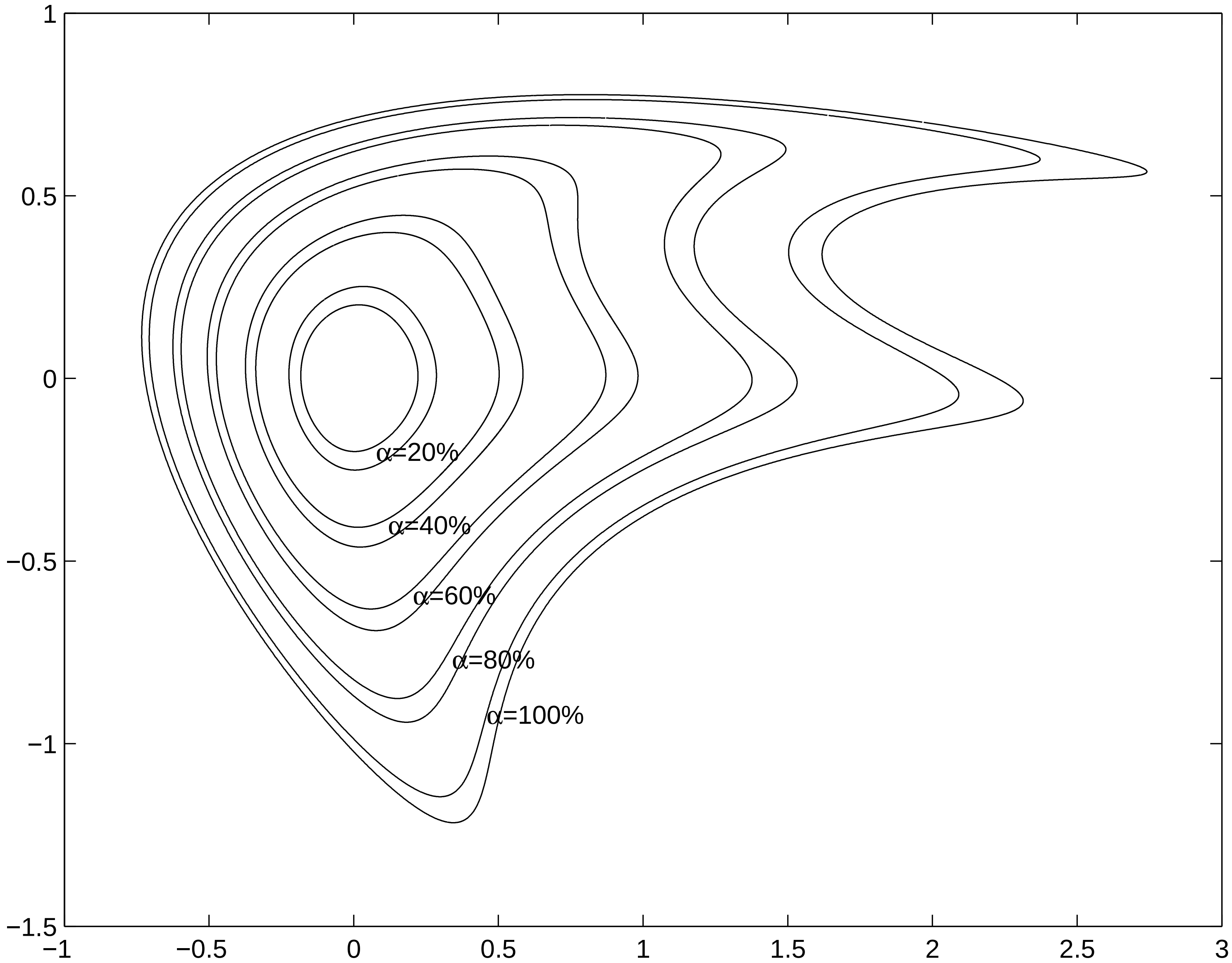
Clouds

allow the representation of incomplete stochastic information in a clearly understandable and computationally attractive way.

They describe the rough shapes of typical samples of various size, without fixing the details of the distribution.

The use of clouds permits a worst case analysis without losing track of important probabilistic information.

All computed probabilities – and hence the resulting designs – are safeguarded against perturbations due to unmodelled (and unavailable) information.



1
0.5
0
−0.5
−1
−1.5
−1
−0.5
0
0.5
1
1.5
2
2.5
3
α=20%
α=40%
α=60%
α=80%
α=100%

The special case of interest for large-scale models is a confocal cloud defined by

- **a continuous potential V which assigns to each scenario x a potential function $V(x)$ defining the shape of the cloud,**
- **a lower probability $\underline{\alpha}(U)$ and an upper probability $\overline{\alpha}(U)$ defining the fuzzy boundary of the cloud, such that, for all U,**

$$\underline{\alpha}(U) \le \Pr(V(x) < U) \le \Pr(V(x) \le U) \le \overline{\alpha}(U).$$

$\underline{\alpha}$ and $\overline{\alpha}$ must be strictly increasing continuous functions of U mapping the range of V to $[0, 1]$.

This corresponds to the level function $\mathbf{x}(\xi)$ defined by
$\underline{x}(\xi) = \underline{\alpha}(V(\xi)), \quad \overline{x}(\xi) = \overline{\alpha}(V(\xi))$**.**

α-Cuts

For a given failure probability ε and $\alpha = 1 - \varepsilon$, the so-called α-cut describes an inner region $\underline{C}_\alpha$ of α-relevant scenarios with $V(x) < \underline{U}_\varepsilon$ and a (generally larger) region $\overline{C}_\alpha$ of α-reasonable scenarios with $V(x) < \overline{U}_\varepsilon$, where

$$\overline{\alpha}(\underline{U}_\varepsilon) = 1 - \varepsilon, \quad \underline{\alpha}(\overline{U}_\varepsilon) = 1 - \varepsilon.$$

The conditions defining the cloud guarantee that $\underline{U}_\varepsilon \le \underline{U}_\varepsilon$, and that there is a region C with $\underline{C}_\alpha \subseteq C \subseteq \overline{C}_\alpha$ containing a fraction of α of all scenarios considered possible.

The potential determines the shape of the cloud. In particular,

- $V(x) = \max_k |x_k - \mu_k|/r_k$ **defines** **rectangular clouds** **(a sort of fuzzy boxes),**
- $V(x) = \|Ax - b\|_2^2$ **defines** **elliptical clouds** **(a sort of fuzzy ellipsoids).**

The construction of appropriate clouds from qualitative or quantitative evidence is a current research topic.

Elliptical clouds are probably appropriate in many applications. However, current numerical experience is restricted to rectangular clouds.

In robust verifiable design, one considers a design <u>safe</u> (at the given admissible failure probability $\varepsilon = 1 - \alpha$) if all α-reasonable scenarios satisfy the response requirements.

One considers a design <u>unsafe</u> if some α-relevent scenario violates the response requirements.

In between there is a grey zone of <u>borderline</u> cases, where more detailed statistical information would be required to fully analyze the safety.

In the simplest case, the safety requirement is given by a single constraint

$$\Pr(s(x) \geq 0) \leq \varepsilon.$$

In this case, the design analysis in terms of a given cloud can be based on the solution of two constraint satisfaction problems involving the safety margin $s(x)$ and the potential $V(x)$.

$$\textbf{(CSP1)} \quad V(x) \leq \overline{U}_\varepsilon, \quad s(x) \geq 0$$
$$\textbf{(CSP2)} \quad V(x) \leq \underline{U}_\varepsilon, \quad s(x) \geq 0$$

	(CSP2) solvable	**(CSP2) inconsistent**
(CSP1) solvable	**unsafe**	**borderline**
(CSP1) inconsistent	$\times$	**safe**

Note that even in the borderline case, useful information for the designer is available:

The solution of (CSP1) may in this case be considered as a weakly unsafe scenario.

A positive decision about safety guarantees that the failure probability is below ε for <u>all</u> possible distributions compatible with the cloud.

Moreover, for any unsafe or borderline case, there is <u>at least one</u> distribution compatible with the cloud for which the failure probability is <u>not</u> below ε.

Note that a <u>single</u> unsafe scenario is considered sufficient ground for <u>rejecting</u> the design.

Has no analogue in probabilistic safety design, but is important in a number of applications:

- **The curiosity of children often makes the most dangerous scenarios most attractive for them.**

- **If the workspace of a robot contains hypersurfaces of singularities, crossing them is dangerous even if they occupy only a negligible part of the volume.**

  The crossing probability may be very large even if the probability of being there is tiny.

As in any worst case analysis, the design resulting from modeling with clouds will be marginally safe for <u>some</u> probability distribution compatible with the information in the cloud.

The design will be oversafe given many other probability distribution compatible with the cloud.

This appears to be the appropriate design goal in all situations where safety is critical.

Monte Carlo simulation

In many cases, simulation models for generating reasonable scenarios are available.

These can be used in Monte Carlo studies to estimate failure probabilities.

The resulting failure probabilities are usually not very robust under variations of the simulation model.

Moreover, Monte Carlo simulations typically require a large number of scenarios, especially when tiny falure probabilities are required.

The evaluation of a design on a single scenario is typically already expensive (solution of a finite element problem), which makes such Monte Carlo studies <u>very</u> expensive.

Simulation in deterministic design

In robust verifiable design, a detailed simulation model can be used to create a large set of representative samples, from which a robust cloud is estimated.

This is the <u>only</u> (and relatively cheap) part involving Monte Carlo techniques, and can be done independent of the evaluation of the design quality or later design optimization.

(Alternatively, a suitable cloud could be determined based on other information, including expert opinion, aggregated information on subsets of variables, etc...)

For design optimization, only the information contained in the cloud is used.

This gives robustness to the design, since it takes account of alternative distributions deviating from the simulation model while being consistent with the derived cloud.

There are also computational advantages since the deterministic evaluation of the quality of a design often needs many fewer scenarios than the determination of the failure probability by Monte Carlo simulation.

Deterministic design optimization

Maximize the design quality

$$\begin{array}{ll} \max & q(c,\varepsilon) \\ \text{s.t.} & F_d(c,\theta) \in \mathbf{F}_d \\ & E(\theta,x,y)=0,\ V(x) \le \overline{U}_\varepsilon \ \Rightarrow\ F_s(c,y) \in \mathbf{F}_s \end{array}$$

The implication constraint can be written as
"$E(\theta,x,y)=0,\ V(x) \le \overline{U}_\varepsilon,\ F_s(c,y) \notin \mathbf{F}_s$ **impossible"**

Thus we have a bilevel optimization problem in which the inner problem involves the negative solution of a CSP. Solution techniques for such problems are a topic of current research.

Conclusions

- **clouds provide a new class of imprecise probability models for robust design**
- **precise and intuitive semantics**
- **based on accessible information only**
- **a poor fit of clouds makes the estimates more conservative but not wrong**
- **large scale problems are tractable via global optimization**
- **optimal design with clouds leads to bilevel optimization problems**

<u>BARON</u> (complete global optimization):
http://archimedes.scs.uiuc.edu/baron/baron.html

<u>SNOBFIT</u> (optimizes noisy, expensive functions):
www.mat.univie.ac.at/∼neum/software/snobfit/

<u>Clouds</u> (code limited probabilistic information):
www.mat.univie.ac.at/∼neum/papers.html#cloud

<u>Surprise</u> (codes qualitative information):
www.mat.univie.ac.at/∼neum/papers.html#fuzzy

<u>The present slides</u> (robust, verifiable design):
www.mat.univie.ac.at/∼neum/ms/uncslides.pdf