Tilburg University

Robust Optimization Using Computer Experiments

Stinstra, E.; den Hertog, D.

Publication date:
2005

No. 2005–90

ROBUST OPTIMIZATION USING COMPUTER EXPERIMENTS

By Erwin Stinstra, Dick den Hertog

July 2005

ISSN 0924-7815

Robust optimization using computer experiments

Erwin Stinstra[*], Dick den Hertog[**]

[*] Center for Quantitative Methods, P.O. Box 414, 5600 AK Eindhoven, The Netherlands, stinstra@cqm.nl
[**] CentER, Tilburg University, P.O. Box 90153, 5000 LE, Tilburg, The Netherlands, D.denHertog@uvt.nl

Abstract

During metamodel-based optimization three types of implicit errors are typically made. The first error is the simulation-model error, which is defined by the difference between reality and the computer model. The second error is the metamodel error, which is defined by the difference between the computer model and the metamodel. The third is the implementation error. This paper presents new ideas on how to cope with these errors during optimization, in such a way that the final solution is robust with respect to these errors. We apply the robust counterpart theory of Ben-Tal and Nemirovsky to the most frequently used metamodels: linear regression and Kriging models. The methods proposed are applied to the design of two parts of the TV tube. The simulation-model errors receive little attention in the literature, while in practice these errors may have a significant impact due to propagation of such errors.

Keywords: computer simulation, robust counterpart, simulation-model error, metamodel error, implementation error.
JEL classification: C61, C15

1 Introduction

Designers are confronted with the task of designing products and processes. Since physical experimentation is often expensive and difficult, computer models are frequently used for simulating physical characteristics. The designer often needs to optimize the design, i.e., to find the best settings for a number of design parameters that influence the critical quality characteristics of the design. A computer simulation run is usually time-consuming and there is a great variety of possible input combinations. For these reasons, a metamodel is constructed, modeling the quality characteristics as explicit functions of the design parameters. Such a metamodel, also called a global approximation model or surrogate model, is obtained by simulating a number of designs. Well-known metamodels are polynomial and Kriging models. Since a metamodel evaluation is much faster than a simulation run, in practice such a metamodel is used instead of the simulation model, to get an insight into the characteristics of the design and to optimize it. A review of metamodeling applications in structural optimization can be found in Barthelemy and Haftka (1993), and in multidisciplinary design optimization in Sobieszczanski-Sobieski and Haftka (1997).

We define three types of implicit errors which are typically made during a design process in which metamodels are used. The first error type is the *simulation-model error*, which is defined by the difference between reality and the computer model. The simulation-model error is a result of the fact that a simulation model by its very nature is a simplification of reality. Thus for example, all kinds of environmental conditions present in reality but constituting only a minor influence on the aspects to be modeled, are usually not taken into account in the computer model. Numerical noise is another source for the simulation-model error. The second error type is the *metamodel error*, which is defined by the difference between the computer model and the metamodel. The third error type is the *implementation error.* In real life, design parameters that can take any continuous value can never be set at the precise value that was predicted as the best setting. Note that there are usually several factors present that influence the critical quality figures of a design, but which cannot be set, for instance, the humidity or temperature during a production process. Errors produced by variation in non-adjustable factors can also be modeled as implementation errors in the context of this paper. These factors should then of course be present in the simulation and metamodels. Figure 1 shows in which stages of the design process the errors are made. All these errors should be taken into account to obtain a robust optimal design.

This paper presents new ideas on how to cope with the simulation-model, the metamodel and the implementation errors during optimization. We will apply the robust counterpart theory of Ben-Tal and Nemirovsky (2002) for the most frequently used metamodels: linear, quadratic, general linear regression, and Kriging models. Ben-Tal and Nemirovsky (2002) approached uncertain parameters in mathematical optimization problems as follows. Based on the knowledge of the uncertainties, a so-called uncertainty region is defined and care is taken that the constraints should hold for each parameter value in the uncertainty region. This in principle is a semi-infinite problem, but Ben-Tal and Nemirovsky (2002) showed that for several important classes of problems and several types of uncertainty regions, the resulting robust optimization problem is tractable. Using this theory, we formulate a number of mathematical programming models that result in robust solutions with respect to the three error sources, and suggest methods to solve these problems.

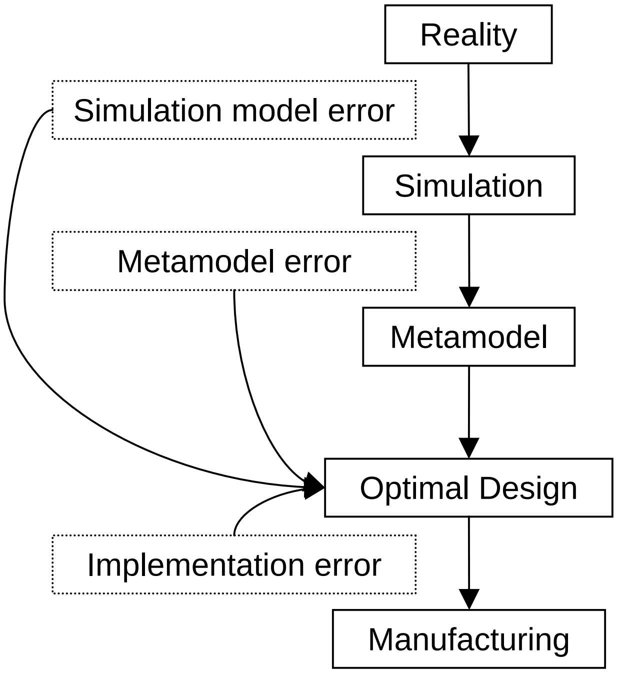

Figure 1. Three types of errors during the metamodelling process.

Most literature on robust design only deals the implementation error. In these papers, the implementation error is described by a known or estimated error distribution (multiplicative/additive). This results in probabilistic constraints. Since such problems are difficult to solve, they are reformulated in terms of the expectations and variances. The goal is to find a setting of the design parameters for which the expected performance of the system is good, and the variance of the system performance is small. See e.g., Mavris, et al. (1996), Sanchez (2000), Putko et al. (2001), and Jin et al. (2003), the six sigma approach of Koch et al. (2004), and the DACE approach of Bates and Wynn (1996), Bates et al. (1999). So far several techniques have been proposed for approximating the response mean and variance, e.g., Taylor's expansion approximations, DOE-based Monte Carlo simulation and the product array approach. See e.g., Koch et al. (1998), Sanchez (2000), Putko et al.

(2001), Doltsinis and Kang (2004). Jin et al. (2003) present a comparison of approximate modeling techniques and optimization formulations suitable for robust optimization with respect to implementation errors. They conclude that the performance of the approximating model is essential to predict the variance and expected constraint infeasibility. Kriging, for example, tends to perform better than polynomial models when robustness is taken into account. A different approach is taken by Gu et al. (2000) and Su and Renaud (1997), who do not work with expectations and variances, but with worst-case uncertainty. This notion is closely related to the robust counterpart notion of Ben-Tal and Nemirovsky (2002), which we also use in this paper.

There are only a few papers that deal with robust optimization with respect to metamodel errors. All these papers concentrate on robust design problems involving real experiments or stochastic simulations. See e.g. Rajagopal and Del Castillo (2005). We do not know of any paper dealing with robust optimization with respect to deterministic simulation-model errors. In Watson (2004), robust optimization methodology is used for robust fitting: given the uncertainty in the measurements, find the best possible model for all possible realizations of the measurement in an uncertainty region. Note that this is a different goal from the one we are trying to reach, which is finding the best settings for the design parameters given the errors that we make in designing computer simulations and model fitting. In our opinion, robust optimization with respect to simulation errors should receive more attention, especially since these errors tend to be quite considerable in practice. This is also illustrated by a TV-tube design problem in this paper.

The outline of this paper is as follows. Section 2 concentrates on notation hat we will use throughout the paper. In Section 3 we sketch the basic robust counterpart method to find robust solutions. Sections 4, 5 and 6 describe mathematical programming methods to cope with the simulation-model, metamodel, and implementation errors, respectively. Section 7 presents two practical cases, in which techniques from Sections 4, 5, and 6 are used. Finally, in Section 8 we draw conclusions and give some recommendations for further research.

2 Notation

In this section, we introduce the notation used in the remainder of this paper.

$n \in \mathrm{N}$	: Number of design parameters
$m \in \mathrm{N}$	: Number of response parameters
$p \in \mathrm{N}$	: Number of computer experiments
$r \in \mathrm{N}$	: Number of constraints on design and response parameters
$t \in \mathrm{N}$	: Number of terms in a linear regression model
$x \in \Re^n$	: Vector of design parameter values
$X \in \Re^{nxn}$	: Diagonal matrix with the elements of x on the main diagonal
$\chi_j \in \Re^n$	: Simulated design parameter settings for the j –th computer experiment
$g : \Re^n \to \Re^t$	: The ‘basis’ functions for a linear regression model
$D \in \Re^{p\times t}$	: Design matrix for a linear regression model
$y^r \in \Re^{m\times p}$	: Real response parameter values
$y^s \in \Re^{m\times p}$	: Simulated response parameter values
$Y_i^s \in \Re^{mxm}$	: Diagonal matrix with the elements of y_i^s on the main diagonal
$f : \Re^n \to \Re^m$	: The ‘real’ functional behavior of the responses (simulation model)
$\hat{f} : \Re^n \to \Re^m$	: The metamodels, i.e. the approximated functional behavior of the simulation model.

Note that

$$D = \begin{bmatrix} g_1(\chi_1) & \dots & g_t(\chi_1) \\ \vdots & & \vdots \\ g_1(\chi_p) & \dots & g_t(\chi_p) \end{bmatrix}, \qquad (1)$$

and the approximated function can be written as

$$\hat{f}_i(x) = \sum_{j=1}^{t} \alpha_{ij} g_j(x), \qquad (2)$$

in which $\alpha_{ij} \in \Re$ are the coefficients. Linear regression models like linear and quadratic functions are usually created by least squares fitting techniques; see e.g. Montgomery et al. (2001). In the general linear regression case, we use the notation defined above. In the case of linear models, we have:

$$\begin{aligned} & t = n + 1 \\ & g_1(x) = 1 \\ & g_i(x) = x_{i-1}, i = 2, \dots, t. \end{aligned} \qquad (3)$$

So, in this case we have

$$D = \begin{bmatrix} 1 & \chi_1^T \\ \vdots & \vdots \\ 1 & \chi_p^T \end{bmatrix}. \qquad (4)$$

The metamodels have the following format for linear models:

$$\hat{f}_i(x) = \sum_{j=1}^{t} \alpha_{ij} g_j(x) = a_i + b_i^T x, \qquad (5)$$

in which $a_i \in \Re, \quad b_i \in \Re^n$. In the case of quadratic models, we first have to define the interaction and quadratic terms that we want to include in the model. Sometimes automatic selection procedures are used like stepwise regression (Montgomery et al. (2001)). The notation of this selection is given by the selection of *j* and *k* in the basis functions for the linear regression:

$$\begin{aligned} & g_1(x) = 1 \\ & g_j(x) = x_{j-1}, j = 2, \dots, n + 1 \\ & g_j(x) = x_k x_l, j = n + 2, \dots, t. \end{aligned} \qquad (6)$$

The quadratic metamodel can be written in the following format:

$$\hat{f}_i(x) = \sum_{j=1}^{t} \alpha_{ij} g_j(x) = a_i + b_i^T x + x^T C_i x, \qquad (7)$$

in which $a_i \in \Re, \quad b_i \in \Re^n$, and $C_i \in \Re^{nxn}$.

The Kriging metamodel has the following format:

$$\hat{f}_i(x) = \delta_{i0} + \sum_{j=1}^{p} \delta_{ij} e^{\sum_{k=1}^{n} -\theta_{ik} \left| x_k - \chi_{jk} \right|^{\rho_{ik}}} , \qquad (8)$$

in which θ_{ik} and ρ_{ik} are the maximum likelihood estimators, and δ_{ij} are constants calculated from the data. Often $\rho_{ik} = 2$ is taken in practice. This Kriging model is interpolating in the design points. Since the Kriging metamodel is also a Best Linear Unbiased Predictor (BLUP), it can also be written as (Sacks et al. (1989))

$$\hat{f}_i(x) = \sum_{j=1}^{p} \beta_{ij}(x) \cdot y_{ij}^s, \qquad (9)$$

in which

$$\beta_i(x) = \frac{1 - e_n{}^T R_i^{-1} r_i(x)}{e_n{}^T R_i^{-1} e_n} R_i^{-1} e_n + R_i^{-1} r_i(x), \qquad (10)$$

and e_n is the n –dimensional all-one vector, and

$$R_i = \begin{bmatrix} R_i(\chi^1, \chi^1) & R_i(\chi^1, \chi^2) & \cdots & R_i(\chi^1, \chi^p) \\ R_i(\chi^2, \chi^1) & R_i(\chi^2, \chi^2) & \cdots & R_i(\chi^1, \chi^p) \\ \vdots & \vdots & \vdots & \vdots \\ R_i(\chi^p, \chi^1) & R_i(\chi^p, \chi^2) & \cdots & R_i(\chi^p, \chi^p) \end{bmatrix} \qquad (11)$$

and

$$r_i(x) = \begin{bmatrix} R_i(\chi^1, x) \\ R_i(\chi^2, x) \\ \vdots \\ R_i(\chi^p, x) \end{bmatrix} \qquad (12)$$

and

$$R_i(w, x) = e^{\sum_{k=1}^{n} -\theta_{ik} \left| w_k - x_k \right|^{\rho_{ik}}} . \qquad (13)$$

Usually, there are a priori known bounds on design parameters which have to be satisfied. We therefore define the *design space* Γ by all combinations of design parameters that satisfy these constraints. We assume that the design space is bounded.

3 Basic robust counterpart method

Since we want to find the best design, i.e. the settings of x in the design space for which the objective f_0 is minimal and all constraints on f_i are satisfied, the initial problem to solve is

$$\min_{x\in\Gamma}\left\{ f_0(x) \middle| f_i(x) \leq \gamma_i, \forall i \right\}, \qquad (14)$$

in which γ is the upper bound for the i-th response. After the metamodels have been constructed, we replace the responses by the approximations and solve the following optimization problem:

$$\min_{x\in\Gamma}\left\{ \hat{f}_0(x) \middle| \hat{f}_i(x) \leq \gamma_i, \forall i \right\}. \qquad (15)$$

Note, that in this optimization problem the simulation-model, metamodel and implementation error are not taken into account. Consequently, the final solution may not be robust. We therefore add some information on these errors to obtain robust solutions. We enforce that the constraints in **(15)** should hold for all reasonably speaking possible scenarios of the errors. This set of scenarios is called the uncertainty region. Note that the user can specify his own uncertainty region, which may be a pessimistic or optimistic one. We will describe the robust counterpart approach in more details for the three possible errors. In the sequel of this paper we will omit $x \in \Gamma$, for simplicity of notation.

Simulation-model error

The simulation-model error is the difference between the reality and the simulation-model prediction. Often, a simulation tool not only gives a predicted value, but also a tolerance, indicating e.g., that a simulated value is somewhere between a lower and an upper bound. We define multiplicative errors and additive errors on the simulation tool results (for each response variable i and for each simulation run j) with the symbols ε_{ij}^m and ε_{ij}^a, respectively, such that

$$y_{ij}^r = \left(1 + \varepsilon_{ij}^m\right) y_{ij}^s + \varepsilon_{ij}^a. \qquad (16)$$

Suppose the uncertainty regions are defined by $\varepsilon_i^m \in U_i^m$ and $\varepsilon_i^a \in U_i^a$. We will treat both box and ellipsoidal constrained uncertainty regions. So, the robust counterpart to solve is:

$$\begin{aligned} &\min_x z \\ s.t.\quad &\hat{f}_0(x) \leq z, \forall y_0^r = y_0^s + Y_0^s \varepsilon_0^m + \varepsilon_0^a,\ \ \varepsilon_0^m \in U_0^m, \varepsilon_0^a \in U_0^a \\ &\hat{f}_i(x) \leq \gamma_i, \forall y_i^r = y_i^s + Y_i^s \varepsilon_i^m + \varepsilon_i^a, \varepsilon_i^m \in U_i^m, \varepsilon_i^a \in U_i^a, \forall i. \end{aligned} \qquad (17)$$

Note that the approximations $\hat{f}_i(x)$ depend on the real response parameter values y^r. In Section 4, we will show how this problem can be reformulated to solvable mathematical programming problems for different metamodel classes and different uncertainty regions.

Metamodel error

The metamodel error is the difference between the simulation-model prediction and the metamodel prediction. Suppose that $error_i^u(x)$ is the upper level of the approximated interval for the real error $error_i(x)$ for the i-th metamodel. Then to get a robust solution we propose to solve the following problem:

$$\min_{z,x}\left\{ z \;\middle|\; z \geq \hat{f}_0(x) + error_i(x),\ \hat{f}_i(x) + error_i(x) \leq \gamma_i, \forall error_i(x) \leq error_i^u(x), \forall i \right\}, \qquad (18)$$

which is equivalent to

$$\min_{z,x}\left\{ z \;\middle|\; z \geq \hat{f}_0(x) + error_0^u(x),\ \hat{f}_i(x) + error_i^u(x) \leq \gamma_i, \forall i \right\}. \qquad (19)$$

We will sketch in Section 5 how to obtain good estimates for these error upper levels.

Implementation error
We consider two types of implementation errors: additive and multiplicative. We define additive implementation errors $\varepsilon \in U$ such that $x_j \mapsto x_j + \varepsilon_j$. Then the robust counterpart of problem **(15)** becomes:

$$\min_{z,x}\left\{ z \;\middle|\; \hat{f}_0(x+\varepsilon) \leq z,\ \hat{f}_i(x+\varepsilon) \leq \gamma_i, \forall i, \forall \varepsilon \in U \right\}. \qquad (20)$$

We define multiplicative implementation errors $\varepsilon \in U$ such that $x_j \mapsto x_j(1+\varepsilon_j)$. Then the robust counterpart of problem **(15)** becomes:

$$\min_{z,x}\left\{ z \;\middle|\; \hat{f}_0(x+X\varepsilon) \leq z,\ \hat{f}_i(x+X\varepsilon) \leq \gamma_i, \forall i, \forall \varepsilon \in U \right\}. \qquad (21)$$

In Section 6, we will show how this problem can be reformulated to solvable mathematical programming problems for different metamodel classes and different types of uncertainty regions.

4 Simulation-model error

The simulation-model error is the difference between the reality and the simulation-model prediction. Often, a simulation tool not only gives a predicted value, but also a tolerance, indicating e.g., that a simulated value is somewhere between a lower and an upper bound. In this section we describe how to obtain optimal solutions which are robust with respect to simulation-model errors for linear, quadratic, linear regression and Kriging models.

4.1 Linear models

In case all metamodels are linear, design optimization problem **(15)** can be rewritten as:

$$\min_{x}\left\{ a_0 + b_0^T x \;\middle|\; a_i + b_i^T x \leq \gamma_i, \forall i \right\} \qquad (22)$$

We discuss two types of uncertainty regions: box and ellipsoidal constrained regions.

Box constrained uncertainty region

We define multiplicative errors and additive errors on the simulation tool results as in **(16)**. Suppose the uncertainty regions are defined by: $-\sigma_{ij}^{m} \leq \varepsilon_{ij}^{m} \leq \sigma_{ij}^{m}$ and $-\sigma_{ij}^{a} \leq \varepsilon_{ij}^{a} \leq \sigma_{ij}^{a}$, in which σ_{ij}^{m} and σ_{ij}^{a} are given positive constants. If we use least squares approximation to fit the linear models, the nominal coefficients (indicated by a bar) can be calculated (see Montgomery et al. (2001)):

$$\begin{bmatrix} \bar{a}_i \\ \bar{b}_i \end{bmatrix} = \left[D^T D\right]^{-1} D^T y_i^s, \qquad \textbf{(23)}$$

where D is the corresponding design matrix **(4)**. Note that in case of interpolation $\left[D^T D\right]^{-1} D$ reduces to D^{-1}, hence our robust counterpart method can also be used for interpolation. Substituting the additive and multiplicative errors in this formula gives

$$\begin{bmatrix} a_i \\ b_i \end{bmatrix} = \left[D^T D\right]^{-1} D^T \left[y_i^s + Y_i^s \varepsilon_i^m + \varepsilon_i^a\right] = \begin{bmatrix} \bar{a}_i \\ \bar{b}_i \end{bmatrix} + \left[D^T D\right]^{-1} D^T \left[Y_i^s \varepsilon_i^m + \varepsilon_i^a\right]. \qquad \textbf{(24)}$$

So, the robust counterpart of **(22)** is:

$$\min_{x,z} \left\{ z \;\middle|\; z \geq a_0 + b_0^T x, a_i + b_i^T x \leq \gamma_i, \forall (a_i, b_i) \in \Omega_i, \forall i \right\}, \qquad \textbf{(25)}$$

with the uncertainty region:

$$\Omega_i = \left\{ (a_i, b_i) \in \Re \times \Re^n \;\middle|\; \begin{bmatrix} a_i \\ b_i \end{bmatrix} = \begin{bmatrix} \bar{a}_i \\ \bar{b}_i \end{bmatrix} + F_i \varepsilon_i^m + G \varepsilon_i^a, -\sigma_i^m \leq \varepsilon_i^m \leq \sigma_i^m, -\sigma_i^a \leq \varepsilon_i^a \leq \sigma_i^a \right\}. \qquad \textbf{(26)}$$

where

$$F_i = \left[D^T D\right]^{-1} D^T Y_i^s \qquad \textbf{(27)}$$

and

$$G = \left[D^T D\right]^{-1} D^T. \qquad \textbf{(28)}$$

Now, we can rewrite each constraint *i* in **(25)** as

$$\begin{aligned} & a_i + b_i^T x \leq \gamma_i, \forall (a_i, b_i) \in \Omega_i \Leftrightarrow \bar{a}_i + \bar{b}_i^T x + \left[F_i \varepsilon_i^m\right]^T \begin{bmatrix} 1 \\ x \end{bmatrix} + \left[G \varepsilon_i^a\right]^T \begin{bmatrix} 1 \\ x \end{bmatrix} \leq \gamma_i, \forall \varepsilon_i^m, \varepsilon_i^a \Leftrightarrow \\ & \bar{a}_i + \bar{b}_i^T x + v_i^T \sigma_i^m + w^T \sigma_i^a \leq \gamma_i, \quad -v_i \leq F_i^T \begin{bmatrix} 1 \\ x \end{bmatrix} \leq v_i, \quad -w \leq G^T \begin{bmatrix} 1 \\ x \end{bmatrix} \leq w. \end{aligned} \qquad \textbf{(29)}$$

So, the robust counterpart problem **(25)** can be written as:

$$
\begin{aligned}
&\min_{x,v,w,z} z \\
&s.t. \quad \bar{a}_0 + \bar{b}_0^{\,T} x + v_0^{\,T} \sigma_0^m + w^T \sigma_0^a \le z \\
&\qquad \bar{a}_i + \bar{b}_i^{\,T} x + v_i^{\,T} \sigma_i^m + w^T \sigma_i^a \le \gamma_i \\
&\qquad -v_i \le F_i^T \begin{bmatrix} 1 \\ x \end{bmatrix} \le v_i\,, \quad \forall i \\
&\qquad -w \le G^T \begin{bmatrix} 1 \\ x \end{bmatrix} \le w, \quad \forall i
\end{aligned}
\qquad \textbf{(30)}
$$

which is again a linear programming problem, but now with $n+(m+1)p+1$ variables and $m+2(m+1)p$ constraints.

Ellipsoidal constrained uncertainty region

Suppose again, that we have an additive error and a multiplicative error in the simulation data as in **(16)**. We now assume that the uncertainty can be defined by an ellipsoid instead of a box. This is the case, e.g., when the simulation output is not an uncertainty interval, but an expected value and a variance. In that case, we have:

$$
\Omega_i = \left\{ (a_i, b_i) \in \Re \times \Re^n \left| \begin{bmatrix} a_i \\ b_i \end{bmatrix} = \begin{bmatrix} \bar{a}_i \\ \bar{b}_i \end{bmatrix} + F_i \varepsilon_i^m + G \varepsilon_i^a, (\varepsilon_i^m)^T Q_i \varepsilon_i^m \le \left(\sigma_i^m\right)^2, (\varepsilon_i^a)^T P_i \varepsilon_i^a \le \left(\sigma_i^a\right)^2 \right. \right\}, \qquad \textbf{(31)}
$$

where Q_i and P_i are positive definite matrices, and F_i and G are as defined in **(27)** and **(28)**, and σ_i^m and σ_i^a are given positive constants. We can rewrite each constraint in **(25)** by maximizing the left-hand side with respect to ε_i^m and ε_i^a (see also Ben-Tal and Nemirovsky (2002)):

$$
\begin{aligned}
& a_i + b_i^T x \le \gamma_i, \forall (a_i, b_i) \in U_i \Leftrightarrow \bar{a}_i + \bar{b}_i^{\,T} x + \begin{bmatrix} 1 & x^T \end{bmatrix} F_i \varepsilon_i^m + \begin{bmatrix} 1 & x^T \end{bmatrix} G \varepsilon_i^a \le \gamma_i, \forall \varepsilon_i^m, \varepsilon_i^a \Leftrightarrow \\
& \bar{a}_i + \bar{b}_i^{\,T} x + \sigma_i^m \sqrt{\begin{bmatrix} 1 & x^T \end{bmatrix} F_i Q_i^{-1} F_i^T \begin{bmatrix} 1 \\ x \end{bmatrix}} + \sigma_i^a \sqrt{\begin{bmatrix} 1 & x^T \end{bmatrix} G P_i^{-1} G^T \begin{bmatrix} 1 \\ x \end{bmatrix}} \le \gamma_i .
\end{aligned}
\qquad \textbf{(32)}
$$

Since $P_i \succ 0, \quad Q_i \succ 0$, and using Sylvester's law of inertia, we have $F_i Q_i^{-1} F_i^T \succ 0$ and $G P_i^{-1} G^T \succ 0$. Hence, we can rewrite the robust counterpart problem **(25)** as the following Second-order Cone Optimization (SOCO) problem:

$$\begin{aligned} &\min_{x,z} z \\ &s.t. \quad \bar{a}_0 + \bar{b}_0{}^T x + \sigma_0^m \sqrt{\begin{bmatrix}1 & x^T\end{bmatrix} F_0 Q_0^{-1} F_0^T \begin{bmatrix}1 \\ x\end{bmatrix}} + \sigma_0^a \sqrt{\begin{bmatrix}1 & x^T\end{bmatrix} G P_0^{-1} G^T \begin{bmatrix}1 \\ x\end{bmatrix}} \le z \\ &\qquad \bar{a}_i + \bar{b}_i{}^T x + \sigma_i^m \sqrt{\begin{bmatrix}1 & x^T\end{bmatrix} F_i Q_i^{-1} F_i^T \begin{bmatrix}1 \\ x\end{bmatrix}} + \sigma_i^a \sqrt{\begin{bmatrix}1 & x^T\end{bmatrix} G P_i^{-1} G^T \begin{bmatrix}1 \\ x\end{bmatrix}} \le \gamma_i, \quad \forall i \end{aligned} \qquad (\ 33\ )$$

with $n+1$ variables and m constraints. Such SOCO problems can be efficiently solved nowadays; see for example Sturm (2002).

4.2 Quadratic models

If we have chosen to use quadratic metamodels, the design optimization problem becomes a (not necessarily convex) quadratic programming problem:

$$\min_{x,z}\left\{ z \mid z \ge a_0 + b_0^T x + x^T C_0 x, a_i + b_i^T x + x^T C_i x \le \gamma_i, \forall i \right\}. \qquad \textbf{(34)}$$

We will not treat quadratic models as a separate case, but as a special case of general linear regression in Section 4.3.

4.3 General linear regression models

For general linear regression models, problem **(15)** can be written as:

$$\min_{x\in\Gamma,z}\left\{ z \;\middle|\; z \ge \sum_j \alpha_{0j} g_j(x), \sum_j \alpha_{ij} g_j(x) \le \gamma_i, \forall i \right\}. \qquad \textbf{(35)}$$

Box constrained uncertainty region

Consider the case that the uncertainty region is a box:

$$\begin{aligned} \Omega_i &= \left\{ \alpha_i \in \Re^p \;\middle|\; \alpha_i = \bar{\alpha}_i + \left[D^T D\right]^{-1} D^T \left[Y_i^s \varepsilon_i^m + \varepsilon_i^a\right], -\sigma_i^m \le \varepsilon_i^m \le \sigma_i^m, -\sigma_i^a \le \varepsilon_i^a \le \sigma_i^a \right\} \\ &= \left\{ \alpha_i \in \Re^p \;\middle|\; \alpha_i = \bar{\alpha}_i + F_i \varepsilon_i^m + G \varepsilon_i^a, -\sigma_i^m \le \varepsilon_i^m \le \sigma_i^m, -\sigma_i^a \le \varepsilon_i^a \le \sigma_i^a \right\}, \end{aligned} \qquad \textbf{(36)}$$

.

in which F_i and G defined as in **(27)** and **(28)** respectively. Hence, the robust counterpart problem of **(35)** becomes:

$$\min_{x,z}\left\{ z \;\middle|\; z \ge \sum_j \alpha_{0j} g_j(x), \sum_j \alpha_{ij} g_j(x) \le \gamma_i, \forall \alpha_i \in \Omega_i \forall i \right\}. \qquad \textbf{(37)}$$

As for the linear case, we can rewrite every constraint of **(37)** as follows:

$$\begin{aligned}
&\alpha_i^T g(x) \le \gamma_i, \forall \alpha_i \in \Omega_i \Leftrightarrow \left(\overline{\alpha}_i + F_i \varepsilon_i^m + G\varepsilon_i^a\right)^T g(x) \le \gamma_i, \forall \varepsilon_i^m, \varepsilon_i^a \Leftrightarrow \\
&\overline{\alpha}_i^T g(x) + g(x)^T F_i \varepsilon_i^m + g(x)^T G\varepsilon_i^a \le \gamma_i, \forall \varepsilon_i^m, \varepsilon_i^a \Leftrightarrow \\
&\overline{\alpha}_i^T g(x) + v_i^T \sigma_i^m + w^T \sigma_i^a \le \gamma_i, -v_i \le F_i^T g(x) \le v_i, \; -w \le G^T g(x) \le w.
\end{aligned} \tag{38}$$

Thus the robust counterpart **(37)** can be rewritten as:

$$\begin{aligned}
&\min_{x,v,w,z} z \\
&s.t. \quad \overline{\alpha}_0^T g(x) + v_0^T \sigma_0^m + w^T \sigma_0^a \le z \\
&\qquad\quad \overline{\alpha}_i^T g(x) + v_i^T \sigma_i^m + w^T \sigma_i^a \le \gamma_i, \quad \forall i \\
&\qquad\quad -v_i \le F_i^T g(x) \le v_i, \quad \forall i \\
&\qquad\quad -w \le G^T g(x) \le w.
\end{aligned} \tag{39}$$

Note that for the linear case this problem is equivalent to problem **(30)**. For the quadratic case this problem is a quadratically constrained quadratic programming problem (not necessarily convex).

Ellipsoidal constrained uncertainty region

Suppose that the uncertainty region is ellipsoidal. Following the same reasoning as in Section 4.1, it is easy to see that we can formulate the uncertainty region for a constraint *i* in **(35)** as:

$$\Omega_i = \left\{\alpha_i \in \Re^p \,\middle|\, \alpha_i = \overline{\alpha}_i + F_i \varepsilon_i^m + G\varepsilon_i^a, \left(\varepsilon_i^m\right)^T Q_i \varepsilon_i^m \le \left(\sigma_i^m\right)^2, \left(\varepsilon_i^a\right)^T P_i \varepsilon_i^a \le \left(\sigma_i^a\right)^2 \right\}. \tag{40}$$

As above, we can reformulate each constraint as follows:

$$\overline{\alpha}_i^T g(x) + \sigma_i^m \sqrt{g(x)^T F_i Q_i^{-1} F_i^T g(x)} + \sigma_i^a \sqrt{g(x)^T G_i P_i^{-1} G_i^T g(x)} \le \gamma_i\,. \tag{41}$$

We can rewrite the robust counterpart of the mathematical programming problem **(35)** as follows:

$$\begin{aligned}
&\min_{x,z} z \\
&s.t. \quad \overline{\alpha}_0^T g(x) + \sigma_0^m \sqrt{g(x)^T F_0 Q_0^{-1} F_0^T g(x)} + \sigma_0^a \sqrt{g(x)^T G_0 P_0^{-1} G_0^T g(x)} \le z \\
&\qquad\quad \overline{\alpha}_i^T g(x) + \sigma_i^m \sqrt{g(x)^T F_i Q_i^{-1} F_i^T g(x)} + \sigma_i^a \sqrt{g(x)^T G_i P_i^{-1} G_i^T g(x)} \le \gamma_i \quad \forall i.
\end{aligned} \tag{42}$$

Unfortunately, in contrast to the linear case, in general we cannot prove any special properties of this problem, not even for the quadratic case. Suppose, for example, that the original problem when quadratic metamodels are used is (by coincidence) a convex problem, then the robust counterpart is likely not to be convex.

4.4 Kriging models

The Kriging metamodel for the i-th response can be written as (see **(43)**)

$$\hat{f}_i(x) = \sum_{j=1}^{p} \beta_{ij}(x) y_{ij}^s \, , \tag{44}$$

in which $\beta_{ij}(x)$ is given in **(10)**. Substituting this model format, we can rewrite the i-th constraint of **(17)** as follows:

$$\beta_i(x)^T y_i^r \leq \gamma_i, \forall y_i^r \in \Omega_i. \tag{45}$$

Box constrained uncertainty region
Suppose the uncertainty region is defined as a box:

$$\Omega_i = \left\{ y_i^r \in \Re^p \middle| y_i^r = y_i^s + Y_i^s \varepsilon_i^m + \varepsilon_i^a, -\sigma_i^m \leq \varepsilon_i^m \leq \sigma_i^m, -\sigma_i^a \leq \varepsilon_i^a \leq \sigma_i^a \right\} \tag{46}$$

Then, we can rewrite the constraint **(45)** as

$$\begin{aligned} &\beta_i(x)^T \left(y_i^s + Y_i^s \varepsilon_i^m + \varepsilon_i^a \right) \leq \gamma_i, \forall \varepsilon_i^m, \varepsilon_i^a \Leftrightarrow \\ &\beta_i(x)^T y_i^s + v_i^T \sigma_i^m + w_i^T \sigma_i^a \leq \gamma_i, -v_i \leq Y_i^s \beta_i(x) \leq v_i, -w_i \leq \beta_i(x) \leq w_i. \end{aligned} \tag{47}$$

The resulting robust counterpart becomes:

$$\begin{aligned} &\min_{x,v,w,z} z \\ &s.t. \quad \beta_0(x)^T y_0^s + v_0^T \sigma_0^m + w_0^T \sigma_0^a \leq z \\ &\qquad \beta_i(x)^T y_i^s + v_i^T \sigma_i^m + w_i^T \sigma_i^a \leq \gamma_i, \quad \forall i \\ &\qquad -v_i \leq Y_i^s \beta_i(x) \leq v_i, \quad \forall i \\ &\qquad -w_i \leq \beta_i(x) \leq w_i, \quad \forall i. \end{aligned} \tag{48}$$

This problem has $n + 2mp + 1$ variables and $m + 4mp$ constraints. However, the $2mp + 1$ extra variables appear only linearly in the constraints and objective.

Ellipsoidal constrained Uncertainty region

Suppose the uncertainty region is defined as:

$$\Omega_i = \left\{ y_i^r \in \Re^p \,\middle|\, y_i^r = y_i^s + Y_i^s \varepsilon_i^m + \varepsilon_i^a,\ \left(\varepsilon_i^m\right)^T Q_i \varepsilon_i^m \le \left(\sigma_i^m\right)^2,\ \left(\varepsilon_i^a\right)^T P_i \varepsilon_i^a \le \left(\sigma_i^a\right)^2 \right\}. \qquad \textbf{(49)}$$

Then, we can rewrite the constraint as

$$\begin{aligned} &\beta_i(x)^T \left(y_i^s + Y_i^s \varepsilon_i^m + \varepsilon_i^a\right) \le \gamma_i, \forall \varepsilon_i^m, \varepsilon_i^a \Leftrightarrow \\ &\beta_i(x)^T y_i^s + \sigma_i^m \sqrt{\beta_i(x)^T Y_i^s Q_i^{-1} Y_i^s \beta_i(x)} + \sigma_i^a \sqrt{\beta_i(x)^T P_i^{-1} \beta_i(x)} \le \gamma_i. \end{aligned} \qquad \textbf{(50)}$$

Note that $Y_i^s Q_i^{-1} Y_i^s$ and P_i^{-1} are positive definite. The robust counterpart problem can be written as:

$$\begin{aligned} &\min_{x,z} z \\ &s.t.\quad \beta_0(x)^T y_0^s + \sigma_0^m \sqrt{\beta_0(x)^T Y_0^s Q_0^{-1} Y_0^s \beta_0(x)} + \sigma_0^a \sqrt{\beta_0(x)^T P_0^{-1} \beta_0(x)} \le z \\ &\qquad\ \ \beta_i(x)^T y_i^s + \sigma_i^m \sqrt{\beta_i(x)^T Y_i^s Q_i^{-1} Y_i^s \beta_i(x)} + \sigma_i^a \sqrt{\beta_i(x)^T P_i^{-1} \beta_i(x)} \le \gamma_i, \quad \forall i. \end{aligned} \qquad \textbf{(51)}$$

As the original problem **(15)**, this problem has to be solved with global optimization methods.

5 Metamodel error

In this section we describe how the metamodel error can be taken into account in **(15)** to obtain a robust solution. Since the treatment for linear and quadratic models is the same as for general linear regression models, we only describe the latter case. For Kriging models we describe another special way.

5.1 General linear regression models

In a stochastic setting, we can estimate the error that we make when using a linear regression model (see Montgomery et al. (2002), Xu and Albin (2003)). However, in the deterministic setting that we are looking at in this paper, this theory makes no sense. In the special case of interpolation, we may use the Kowaleckski-Ciarlet-Wagschal formula to obtain an upper bound for the interpolation error (see Waldron (1998)). In the more general regression case, cross-validation or bootstrap statistics may be used to estimate the mean-squared-error of the metamodel (see Efron and Tibshirani (1993), Kleijnen and Van Beers (2004)). Disadvantage of this method is that it predicts non-zero errors in the simulation points themselves, even when an interpolating model is used. In fact, we observed that the error may even be predicted to be the largest in these points, when of course it should be exactly zero. Rajagopal and Del Castillo (2005) describe a Bayesian approach to obtain an expression for the model-averaged posterior predictive density, which can also be used as an approximation for the metamodel error. Given the expressions for the metamodel error, we have to solve **(19)**.

5.2 Kriging models

In the literature it is described how to estimate the Mean Squared Error (MSE) for a Kriging model. Then, according to the Kriging variance formula (Sachs et al. (1989), Lophaven et al. (2002)), the error that is made by using a Kriging model $\hat{f}(x)$ can be calculated by:

$$MSE\left[\hat{f}(x)\right]=\sigma^2\left(1+u^T(x)\left(e_n{}^T R^{-1}e_n\right)^{-1}u(x)-r^T(x)R^{-1}r(x)\right), \qquad \textbf{(52)}$$

where σ^2 is the process variance, and $u(x)=e_n{}^T R^{-1}r(x)-e_n$.

We propose to use the upper bound of the $(1-\omega)$ confidence interval to obtain a robust solution:

$$\min_{z,x}\left\{\ z\ \middle|\ z\geq\ \hat{f}_0(x)+t_\omega\sqrt{MSE(\hat{f}_0(x))},\ \hat{f}_i(x)+t_\omega\sqrt{MSE(\hat{f}_i(x))}\leq\gamma_i,\forall i\right\}, \qquad \textbf{(53)}$$

where t_ω is the $(1-\omega)$ quantile of the t-distribution. These confidence intervals are also used by Jones (2001) to obtain global optimal solutions.

6 Implementation error

In this section we show how the implementation error can be taken into account while optimizing. For each class of models we will formulate solvable robust counterpart problems.

6.1 Linear models

We consider four cases: additive and multiplicative errors, with box or ellipsoidal constrained uncertainty regions. In case all models are linear, design optimization problem **(20)**, the additive case, can be defined as:

$$\min_{z,x}\left\{\ z\ \middle|\ a_0+b_0^T x+b_0^T\varepsilon\leq z, a_i+b_i^T x+b_i^T\varepsilon\leq\gamma_i,\forall i,\forall\varepsilon\in U\right\} \qquad \textbf{(54)}$$

and problem **(21)**, the multiplicative case, can be defined as

$$\min_{z,x}\left\{\ z\ \middle|\ a_0+b_0^T x+b_0^T X\varepsilon\leq z, a_i+b_i^T x+b_i^T X\varepsilon\leq\gamma_i,\forall i,\forall\varepsilon\in U\right\}. \qquad \textbf{(55)}$$

Box constrained uncertainty with additive errors

We consider additive errors, and suppose the uncertainty region is defined by the following box constraints: $-\sigma_j\leq\varepsilon_j\leq\sigma_j$. Then it is easy to see that the robust counterpart problem **(54)** becomes the following linear programming problem:

$$\min_{x,z}\left\{\ z\ \middle|\ a_0+b_0^T x\leq z-|\,b_0\,|^T\ \sigma,\ \ a_i+b_i^T x\leq\gamma_i-|\,b_i\,|^T\ \sigma\ \ ,\forall i\right\} \qquad \textbf{(56)}$$

in which $|\,b_i\,|$ is the component wise absolute value of the vector b_i.

Box constrained uncertainty with multiplicative errors

We consider multiplicative errors and assume that the uncertainty region is defined by the following box constraints: $-\sigma_j \le \varepsilon_j \le \sigma_j$. Then it is easy to verify that the robust counterpart problem **(55)** becomes the following linear programming problem:

$$\begin{aligned} &\min_{x,v,z} \; z \\ s.t.\quad & a_0 + b_0^T x + (v_0)^T \sigma \quad \le z \\ & a_i + b_i^T x + (v_i)^T \sigma \quad \le \gamma_i, \quad \forall i \\ & -v_i \le X b_i \le v_i, \quad \forall i. \end{aligned} \tag{57}$$

This problem is again a linear programming problem.

Ellipsoidal constrained uncertainty with additive errors
Since box constrained uncertainty is often too pessimistic, we therefore analyze ellipsoidal constrained uncertainty. The motivation is as follows. Suppose that ε_i is a random variable with mean 0 and variance σ_i^2, then it is unlikely that all random variables will have values close to the extremes. We therefore restrict the uncertainty region to the confidence interval: $\varepsilon^T Q \varepsilon \le \sigma^2$, in which Q is a positive definite matrix. Now it is easy to verify that the robust counterpart (for additive errors) becomes the following linear programming problem:

$$\begin{aligned} &\min_{x,z} \; z \\ s.t.\quad & a_0 + b_0^T x \le z - \sigma \sqrt{b_0^T Q^{-1} b_0} \\ & a_i + b_i^T x \le \gamma_i - \sigma \sqrt{b_i^T Q^{-1} b_i}, \quad \forall i. \end{aligned} \tag{58}$$

Note that the resulting problem is again a linear programming problem. Another derivation for the same robust counterpart can be obtained by adding confidence intervals for the linear functions. To be more precise, for the variance we have $\operatorname{var}(b_i^T(x+\varepsilon)) = b_i^T \operatorname{cov}(\varepsilon) b_i$, and the upper confidence intervals for the linear functions are $b_i^T x + t_{\varpi} \sqrt{b_i^T \operatorname{cov}(\varepsilon) b_i}$.

Ellipsoidal constrained uncertainty with multiplicative errors

For this case it can easily be shown that the robust counter part is the following optimization problem:

$$\begin{aligned} &\min_{x,z} z \\ s.t.\quad & a_0 + b_0^T x + \sigma \sqrt{b_0^T X Q^{-1} X b_0} \le z \\ & a_i + b_i^T x + \sigma \sqrt{b_i^T X Q^{-1} X b_i} \le \gamma_i, \quad \forall i, \end{aligned} \tag{59}$$

which is a second-order cone optimization problem. Such problems can efficiently be solved nowadays.

6.2 Quadratic models

Box constrained uncertainty with additive errors

In the case of box constrained uncertainty and additive errors the robust counterpart problem **(20)** becomes the following problem:

$$\begin{aligned}&\min_{x,z} z\\ &s.t.\quad a_0 + b_0^T x + x^T C_0 x + (b_0 \ + 2C_0 x)^T \varepsilon + \varepsilon^T C_0 \varepsilon \leq z_i, \forall \varepsilon \in U.\\ &\qquad\ a_i + b_i^T x + x^T C_i x + (b_i \ + 2C_i x)^T \varepsilon + \varepsilon^T C_i \varepsilon \leq \gamma_i, \forall i \forall \varepsilon \in U.\end{aligned} \qquad \textbf{(60)}$$

Note that normally speaking C_i is not positive semi-definite. In fact in Den Hertog et al. (2002) it is shown that even if the underlying true function is convex, then the least squares quadratic approximation is not necessarily convex. They also showed how to obtain a quadratic regression model which is convexity preserving. One can show that the maximum of a convex quadratic function over a box constrained region attains its maximum in one of the corners. So, when all C_i are positive semi-definite, then problem **(60)** can be formulated as:

$$\begin{aligned}&\min_{x,z} z\\ &s.t.\quad a_0 + b_0^T x + x^T C_0 x + (b_0 \ + 2C_0 x)^T \varepsilon + \varepsilon^T C_0 \varepsilon \leq z, \forall \varepsilon \in V\\ &\qquad\ a_i + b_i^T x + x^T C_i x + (b_i \ + 2C_i x)^T \varepsilon + \varepsilon^T C_i \varepsilon \leq \gamma_i, \forall i \forall \varepsilon \in V,\end{aligned} \qquad \textbf{(61)}$$

in which V is the set of vertices of the box constrained region. Note that although the number of variables in simulation-based optimization is often restricted, still the number of corners $|V| = 2^n$ can be very large. Formulation **(61)** is only useful for very small values of n. Suppose now that the C_i are not positive semi-definite. Then, a practical way of dealing with **(60)** is to ignore the second-order term for ε, since this is normally speaking very small. Then, **(60)** becomes the following quadratically constrained problem:

$$\begin{aligned}&\min_{x,v,z} z\\ &s.t.\quad a_0 + b_0^T x + x^T C_0 x + v_0{}^T \sigma \leq z\\ &\qquad\ a_i + b_i^T x + x^T C_i x + v_i{}^T \sigma \leq \gamma_i, \forall i\\ &\qquad -v_i \leq b_i \ + 2C_i x \leq v_i, \forall i.\end{aligned} \qquad \textbf{(62)}$$

We can also get accurate lower and upper bounds for the optimal value. It can easily be seen that the optimal value of the following problem is a lower bound for the optimal value of **(60)**:

$$\begin{aligned}
&\min_{x,v,z} z \\
&s.t. \quad a_0 + b_0^T x + x^T C_0 x + v_0{}^T \sigma + \tau_0^{\min} \le z \\
&\qquad a_0 + b_0^T x + x^T C_0 x + \kappa_0^{\min} \le z \\
&\qquad a_i + b_i^T x + x^T C_i x + v_i{}^T \sigma + \tau_i^{\min} \le \gamma_i, \forall i \\
&\qquad a_i + b_i^T x + x^T C_i x + \kappa_i^{\min} \le \gamma_i, \forall i \\
&\quad -v_i \le b_i + 2C_i x \le v_i, \forall i,
\end{aligned} \qquad \textbf{(63)}$$

in which $\tau_i^{\min} = \min_{\varepsilon \in U} (\varepsilon^T C_i \varepsilon)$ and $\kappa_i^{\min} = \min_{\varepsilon \in U, x} ((b_i + 2C_i x)^T \varepsilon + \varepsilon^T C_i \varepsilon)$. On the other hand, the optimal value of the following problem is an upper bound for the optimal value of **(60)**:

$$\begin{aligned}
&\min_{x,v,z} z \\
&s.t. \quad a_0 + b_0^T x + x^T C_0 x + v_0{}^T \sigma + \tau_0^{\max} \le z, \\
&\qquad a_0 + b_0^T x + x^T C_0 x + \kappa_0^{\max} \le z, \\
&\qquad a_i + b_i^T x + x^T C_i x + v_i{}^T \sigma + \tau_i^{\max} \le \gamma_i, \forall i \\
&\qquad a_i + b_i^T x + x^T C_i x + \kappa_i^{\max} \le \gamma_i, \forall i \\
&\quad -v_i \le b_i + 2C_i x \le v_i, \forall i,
\end{aligned} \qquad \textbf{(64)}$$

in which $\tau_i^{\max} = \max_{\varepsilon \in U} (\varepsilon^T C_i \varepsilon)$ and $\kappa_i^{\max} = \max_{\varepsilon \in U, x} ((b_i + 2C_i x)^T \varepsilon + \varepsilon^T C_i \varepsilon)$.

Box constrained uncertainty with multiplicative errors

In the case of multiplicative errors, the robust counterpart problem **(21)** becomes the following problem:

$$\begin{aligned}
&\min_{x,z} z \\
&s.t. \quad a_0 + b_0^T x + x^T C_0 x + (Xb_0 + 2XC_0 x)^T \varepsilon + \varepsilon^T XC_0 X \varepsilon \le z, \forall \varepsilon \in U \\
&\qquad a_i + b_i^T x + x^T C_i x + (Xb_i + 2XC_i x)^T \varepsilon + \varepsilon^T XC_i X \varepsilon \le \gamma_i, \forall i \forall \varepsilon \in U.
\end{aligned} \qquad \textbf{(65)}$$

When all C_i are positive semi-definite (and hence $XC_i X$ are positive semi-definite), this can be rewritten as in the previous case:

$$\begin{aligned}
&\min_{x,z} z \\
&s.t. \quad a_0 + b_0^T x + x^T C_0 x + (Xb_0 + 2XC_0 x)^T \varepsilon + \varepsilon^T XC_0 X \varepsilon \le z, \forall \varepsilon \in V \\
&\qquad a_i + b_i^T x + x^T C_i x + (Xb_i + 2XC_i x)^T \varepsilon + \varepsilon^T XC_i X \varepsilon \le \gamma_i, \forall i \forall \varepsilon \in V.
\end{aligned} \qquad \textbf{(66)}$$

in which V is the set of vertices of the box constrained region. This is again a quadratic programming problem. When n is too large or when C_i is not positive semi-definite, we can again neglect the second-order term for ε, and then **(65)** becomes

$$\begin{aligned} &\min_{x,v,z} z \\ &s.t. \quad a_0 + b_0^T x + x^T C_0 x + v_0{}^T \sigma \le z \\ &\qquad\;\; a_i + b_i^T x + x^T C_i x + v_i{}^T \sigma \le \gamma_i, \forall i. \\ &\qquad\;\; -v_i \le Xb_i \; + 2XC_i x \le v_i, \forall i, \end{aligned} \qquad \textbf{(67)}$$

which is again a quadratic programming problem. Of course, we can also formulate equivalent problems as **(63)** and **(64)** to obtain lower and upper bounds for the optimal value of **(65)**.

Ellipsoidal constrained uncertainty with additive errors

In this case the robust counterpart problem becomes **(60)**, but now the uncertainty region is the ellipsoid $\varepsilon^T Q \varepsilon \le \sigma^2$. By ignoring the second-order terms it can easily be shown that the problem becomes the following second-order cone optimization problem:

$$\begin{aligned} &\min_{x,z} z \\ &s.t. \quad a_0 + b_0^T x + x^T C_0 x + \sigma\sqrt{(b_0 + 2C_0 x)^T Q^{-1} (b_0 + 2C_0 x)} \le z \\ &\qquad\;\; a_i + b_i^T x + x^T C_i x + \sigma\sqrt{(b_i + 2C_i x)^T Q^{-1} (b_i + 2C_i x)} \le \gamma_i, \forall i. \end{aligned} \qquad \textbf{(68)}$$

Of course, we can also formulate equivalent problems as **(63)** and **(64)** to obtain lower and upper bounds for the optimal value of **(60)**.

Ellipsoidal constrained uncertainty with multiplicative errors

In this case the problem becomes **(60)**, but now the uncertainty region is the ellipsoid $\varepsilon^T Q \varepsilon \le \sigma^2$. By ignoring the second-order terms it can easily be shown that the problem becomes:

$$\begin{aligned} &\min_{x,z} z \\ &s.t. \quad a_0 + b_0^T x + x^T C_0 x + \sigma\sqrt{(b_0 + 2C_0 x)^T XQ^{-1} X (b_0 + 2C_0 x)} \le z \\ &\qquad\;\; a_i + b_i^T x + x^T C_i x + \sigma\sqrt{(b_i + 2C_i x)^T XQ^{-1} X (b_i + 2C_i x)} \le \gamma_i, \forall i. \end{aligned} \qquad \textbf{(69)}$$

Note that this problem is not a second-order cone optimization problem, and even if the original problem is convex, this problem may be non convex. Of course, we can also formulate equivalent problems as **(63)** and **(64)** to obtain lower and upper bounds for the optimal value of **(60)**.

6.3 General linear regression models

Again, we consider four cases: additive and multiplicative errors, with box or ellipsoidal constrained uncertainty regions. In case all models are general linear regression models, problem **(20)** is very difficult to solve. Since we may assume that the error is relatively small, we therefore linearize with respect to ε. For the additive case **(20)** we obtain:

$$\min_{z,x}\left\{ z \;\middle|\; \hat{f}_0(x)+\varepsilon^T\nabla\hat{f}_0(x)\le z,\ \hat{f}_i(x)+\varepsilon^T\nabla\hat{f}_i(x)\le\gamma_i, \forall i, \forall\varepsilon\in U\right\}, \qquad \textbf{(70)}$$

in which $\nabla\hat{f}_i(x)=\sum_{j=1}^{t}\alpha_{ij}\nabla g_j(x)$. For the multiplicative case **(21)** we get

$$\min_{z,x}\left\{ z \;\middle|\; \hat{f}_0(x)+\varepsilon^T X\nabla\hat{f}_0(x)\le z,\ \hat{f}_i(x)+\varepsilon^T X\nabla\hat{f}_i(x)\le\gamma_i, \forall i, \forall\varepsilon\in U\right\}. \qquad \textbf{(71)}$$

Box constrained uncertainty with additive errors

We consider additive errors, and suppose the uncertainty region is defined by the following box constraints: $-\sigma_j\le\varepsilon_j\le\sigma_j$. Then it is easy to see that the robust counterpart problem **(70)** becomes the following linear programming problem:

$$\min_{z,x}\left\{ z \;\middle|\; \hat{f}_0(x)+\sigma^T\,|\nabla\hat{f}_0(x)|\le z,\ \hat{f}_i(x)+\sigma^T\,|\nabla\hat{f}_i(x)|\le\gamma_i, \forall i, \forall\varepsilon\in U\right\}, \qquad \textbf{(72)}$$

in which $|\nabla\hat{f}_i(x)|$ is the component wise absolute value of the vector $\nabla\hat{f}_i(x)$. This problem can be rewritten as:

$$\begin{aligned}
&\min_{x,v,z}\ z\\
&s.t.\quad \hat{f}_0(x)+(v_0)^T\sigma\le z\\
&\qquad\ \ \hat{f}_i(x)+(v_i)^T\sigma\le\gamma_i,\quad\forall i\\
&\qquad\ \ -v_i\le\nabla\hat{f}_i(x)\le v_i,\quad\forall i.
\end{aligned} \qquad \textbf{(73)}$$

Box constrained uncertainty with multiplicative errors

We consider multiplicative errors and assume that the uncertainty region is defined by the following box constraints: $-\sigma_j\le\varepsilon_j\le\sigma_j$. Then it is easy to verify that problem **(71)** becomes the following nonlinear programming problem:

$$\begin{aligned} &\min_{x,v,z} z \\ &s.t. \quad \hat{f}_0(x) + (v_0)^T \sigma \le z \\ &\qquad\ \ \hat{f}_i(x) + (v_i)^T \sigma \le \gamma_i, \quad \forall i \\ &\qquad\ -v_i \le X\nabla\hat{f}_i(x) \le v_i, \quad \forall i. \end{aligned} \tag{74}$$

Ellipsoidal constrained uncertainty with additive errors
We restrict the uncertainty region to the confidence interval: $\varepsilon^T Q \varepsilon \le \sigma^2$, in which Q is a positive definite matrix. Now it is easy to verify that problem **(70)** becomes the following nonlinear programming problem:

$$\begin{aligned} &\min_{x,z} z \\ &s.t. \quad \hat{f}_0(x) + \sigma\sqrt{\nabla\hat{f}_0(x)^T Q^{-1} \nabla\hat{f}_0(x)} \le z, \\ &\qquad\ \ \hat{f}_i(x) + \sigma\sqrt{\nabla\hat{f}_i(x)^T Q^{-1} \nabla\hat{f}_i(x)} \le \gamma_i, \quad \forall i. \end{aligned} \tag{75}$$

Ellipsoidal constrained uncertainty with multiplicative errors
For this case it can easily be shown that problem **(71)** becomes the following nonlinear optimization problem:

$$\begin{aligned} &\min_{x,z} z \\ &s.t. \quad \hat{f}_0(x) + \sigma\sqrt{\nabla\hat{f}_0(x)^T XQ^{-1} X\nabla\hat{f}_0(x)} \le z, \\ &\qquad\ \ \hat{f}_i(x) + \sigma\sqrt{\nabla\hat{f}_i(x)^T XQ^{-1} X\nabla\hat{f}_i(x)} \le \gamma_i, \quad \forall i. \end{aligned} \tag{76}$$

6.4 Kriging models

We can treat Kriging metamodels **(77)** in the same way as we described for linear regression models. Note that for Kriging functions we have

$$\frac{\partial \hat{f}_i(x)}{\partial x_l} = -2\theta_{il} \sum_{j=1}^{p} (x_l - \chi_{jl}) \delta_{ij} e^{\sum_{k=1}^{n} -\theta_{ik} \left| x_k - \chi_{jk} \right|^2} . \tag{78}$$

7 Application: robust optimization of several parts of the TV tube

In this section we will apply some of the robust optimization techniques of the previous sections to the design of several parts of the TV tube. The metamodel approach has been successfully applied to the design of several parts of the TV tube; see Den Hertog and Stehouwer (2002) for a detailed treatment on these projects. In this section we apply the robust counterpart methodology to get robust solutions.

7.1 Robust optimization of furnace profile

The first application concerns the optimization of the enameling process. Given the geometry of the screen and the cone, the thermal stresses during the enameling process can be influenced by imposing the right oven temperature profile on the screen and cone. Figure 2 gives an example of a temperature profile. When the stresses on some specified critical area are too high, there is much scrap due to implosions. To minimize the scrap, the designer is interested in the optimal temperature profile such that:

- The temperature values are between certain specified lower and upper bounds.
- The temperature differences between nearby temperature locations on the screen are physically realizable, i.e., not too big.
- The maximal stress at the specified critical area is minimized.

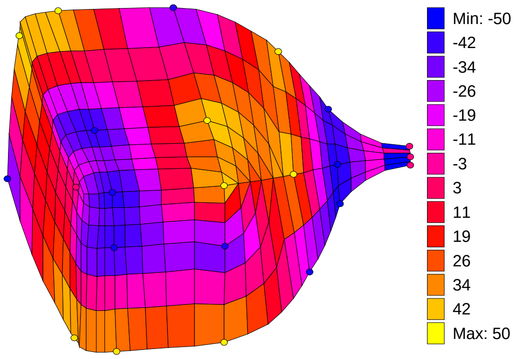

Figure 2. An example of a temperature profile.

The temperature on 23 locations on the screen defines the profile. A FEM model is developed to calculate the 210 thermal stresses for a given temperature profile. A typical simulation run takes several hours. The temperature profile optimization problem is to find values for the 23 temperatures such that the three requirements mentioned above are fulfilled. We refer to Den Hertog and Stehouwer (2002) for a more detailed treatment of this project. The optimization problem can be described as:

$$\begin{array}{ll} \min s_{\max} & \\ s.t. & s_{\max} \geq s_i(x), \qquad \forall i \\ & l \leq x \leq u \\ & -\Delta T_{\max} \leq Ax \leq \Delta T_{\max}, \end{array} \tag{79}$$

with variables $s_{\max}$ representing the maximum stress, $s_i(x)$ representing the stress on node *i*, *x* representing the vector of temperatures. The vectors *l* and *u* denote the lower and upper bounds, respectively. The parameter $\Delta T_{\max}$ represents the maximal allowed temperature differences; the last linear constraints are to enforce that the temperatures on nodes that are close to each other should not differ more than $\Delta T_{\max}$ in an absolute sense.

As described in Den Hertog and Stehouwer (2002) the stresses are modeled as linear functions in x. Substituting the regression model and rewriting the mathematical program in standard form gives:

$$\begin{aligned} &\min s_{\max} \\ &s.t. \quad a_i + b_i^T x - s_{\max} \leq 0, \qquad \forall i \\ &\qquad l \leq x \leq u \\ &\qquad -\Delta T_{\max} \leq Ax \leq \Delta T_{\max}. \end{aligned} \qquad (80)$$

In fact, this problem was solved in Den Hertog and Stehouwer (2002) to obtain big improvements. We now assume that the simulation tool gives a result within accuracy of 4% and we look at the robust counterpart of this problem. So, we have a multiplicative error with $\sigma_i^m = 0.04$ for each simulation outcome. So, we suppose that the worst case relative errors for all response parameters (stresses) and all experiments are equal. Then, the robust version of the furnace optimization problem becomes (see **(30)**):

$$\begin{aligned} &\min s_{\max} \\ &s.t. \quad a_i + b_i^T x - s_{\max} + y_i^T \sigma_i^m \leq 0, \qquad \forall i \\ &\qquad l \leq x \leq u \\ &\qquad -\Delta T_{\max} \leq Ax \leq \Delta T_{\max} \\ &\qquad -y_i \leq F_i^T \begin{bmatrix} 1 \\ x \end{bmatrix} \leq y_i, \qquad \forall i \end{aligned} \qquad (81)$$

in which $F_i = \left[D^T D\right]^{-1} D^T Y_i^s$. We have solved this linear programming problem, and we have compared this robust solution with the nominal solution, i.e., the solution of (80)**.** In Figure 3 the effect of robust optimization is visualized. We have simulated a number of error realizations. The errors were assumed to be uniformly distributed and independent. Given a realization of the possible simulation-model outcome, the effect on the metamodels has been calculated. Using these metamodels, the maximum stress is predicted in the nominal optimum and in the robust optimum. The left graph in Figure 3 shows the distribution of the maximum stress given the uncertainty in the simulation tool output in the nominal optimum; the right graphs shows the situation in the robust optimum. For the robust optimum, both the mean and the variance are significantly smaller than for the nominal optimum. This proves the applicability of our approach. The optimal value for the nominal solution is 14.16. Note that all objective values in the left graph in Figure 3 are worse. This is due to the fact that the objective is the maximum of many stresses, and the fact that the error $\sigma_i^m = 0.04$ is multiplied by elements of the matrix F_i (propagation of errors).

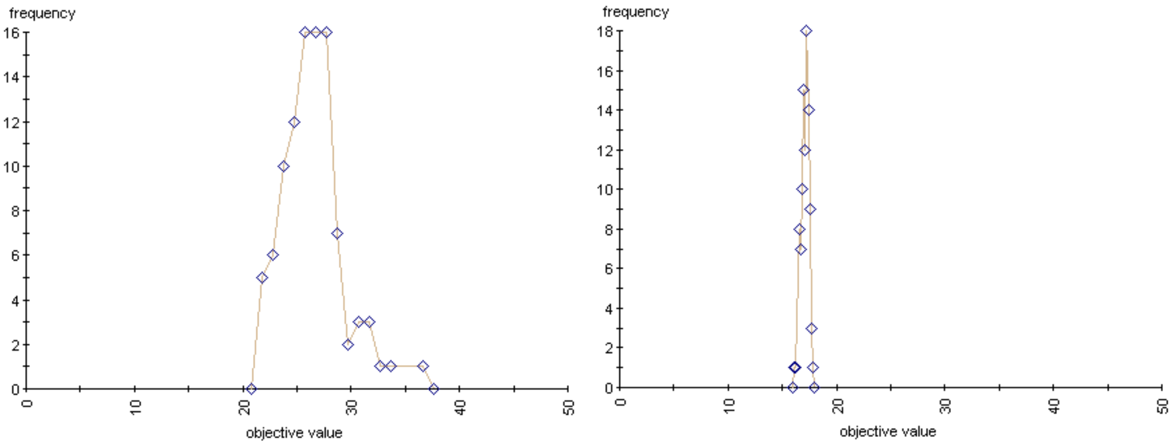

Figure 3:The effect of uniformly distributed noise (4%) on the objective value (100 samples); left in the nominal solution of (80), right in the robust solution of (81).

Ellipsoidal uncertainty region

Suppose the uncertainty is defined as an ellipsoid. This is a more natural situation, since it is usually very unlikely that all errors in the simulations would take an extreme value. Now, we assume the errors to be normally distributed and independent. Suppose we would like to take 99% of all possible situations into account, then the uncertainty region can be defined as:

$$\Omega_i = \left\{ (a_i, b_i) \in \Re \times \Re^n \left| \begin{bmatrix} a_i \\ b_i \end{bmatrix} = \begin{bmatrix} \overline{a}_i \\ \overline{b}_i \end{bmatrix} + F_i \varepsilon_i^m ,\ (\varepsilon_i^m)^T \varepsilon_i^m \le (3\sigma_i^m)^2 \right. \right\}, \qquad \textbf{(82)}$$

This region is much less conservative than the previous one. Then, the robust version of the furnace optimization problem becomes (see **(33)**):

$$\begin{aligned} & \min s_{\max} \\ & s.t. \quad a_i + b_i^T x - s_{\max} + 3\sigma_i^m \sqrt{\begin{bmatrix} 1 & x^T \end{bmatrix} F_i F_i^T \begin{bmatrix} 1 \\ x \end{bmatrix}} \le 0, \qquad \forall i \\ & \qquad l \le x \le u \\ & \qquad -\Delta T_{\max} \le Ax \le \Delta T_{\max}, \end{aligned} \qquad \textbf{(83)}$$

in which $F_i = \left[D^T D \right]^{-1} D^T Y_i^s$. We have solved this second-order cone problem, and we have compared this robust solution with the nominal solution, i.e., the solution of (80).

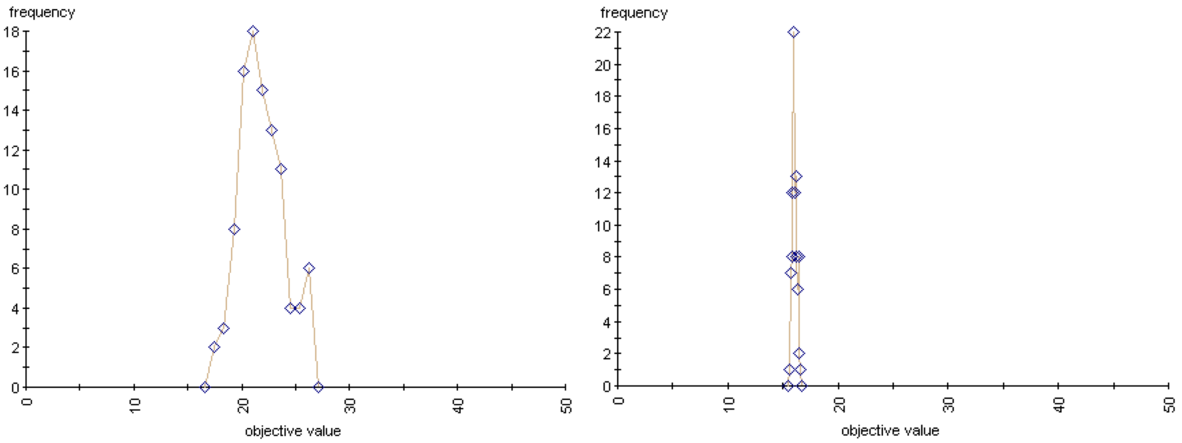

Figure 4: The effect of normally distributed noise (4%) on the objective value (100 samples); left in the nomimal solution of (80), right in the robust solution of (83).

In Figure 4 the effect of robust optimization is visualized. We have simulated a number of error realizations, as in the box constaint case. However, here the errors were assumed to be normally distributed and independent. The left graph in Figure 4 shows the distribution of the maximum stress given the uncertainty in the simulation tool output in the nominal optimum; the right graphs shows the situation in the robust optimum. Again, for the robust optimum, both the mean and the variance are significantly smaller than for the nominal optimum. This proves the applicability of our approach.

7.2 Case 2: robust optimization of the shadow mask

The second application concerns the optimization of the shadow mask design. The essence of shadow mask design is to find the right mask geometry; see the above picture of a shadow mask. In this case, the mask has a doubly curved surface with a geometry that is described by three parameters η_1, η_2, and η_3. The mask surface is given by the following polynomial:

$$z(x,y) = \eta_1 x^2 + \eta_2 y^2 + \eta_3 x^2 y^2 . \qquad \textbf{(84)}$$

In Figure 5 the mask surface for some combination of η_1, η_2, and η_3 is shown.

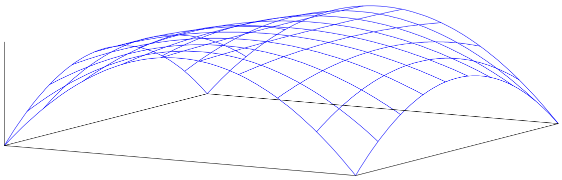

Figure 5. Example of the geometry of a shadow mask

The mask geometry should be optimized with respect to the following three aspects:

- The mask should fit well within the given screen. This means that the given screen geometry restricts the set of possible mask geometries. Table 1 gives upper and lower bounds for each geometry parameter.

- Moreover, the shadow mask height given by $\eta_1 + \eta_2 + \eta_3$ must be between 20 and 32 mm, that is,

$$20 \leq \eta_1 + \eta_2 + \eta_3 \leq 32 \qquad \textbf{(85)}$$

- Drop test – High accelerations due to impacts may cause a shadow mask to buckle. The buckling load is defined as the minimal force (Newton) on the convex side of the mask such that the mask buckles. To optimize its drop resistance, the buckling load of a mask should be as high as possible.
- Picture quality – The third aspect concerns the picture quality of the final TV tube. Mask displacements due to heating of the mask negatively influence the picture quality of a tube. Therefore, the maximal mask displacement under heating should be minimal. In this case, the picture quality is expressed as the ratio of the predefined allowed displacement and the actual displacement. This picture quality ratio has obviously to be larger than 1.0.

Geometry parameter	**Lower bound**	**Upper bound**	**Std. deviation**
η_1	9.40	17.40	0.1
η_2	5.17	11.17	0.1
η_3	1.79	7.79	0.1

Table 1. Bounds and distribution on geometry parameters.

A FEM model of the mask is used to evaluate mask designs. Mask displacements and buckling loads are calculated using a thermal analysis and a linear Eigenvalue analysis, respectively. Computation time of a typical simulation run is about one hour of CPU time on a workstation. We note that these two analyses are done by two different groups of DCE. The mask geometry optimization problem is to find values for the five mask geometry parameters such that the three requirements mentioned above are fulfilled. We refer to Stehouwer and Den Hertog (1999) and Den Hertog and Stehouwer (2002) for more detailed treatments of this project.

When an optimal design is produced, an implementation error occurs. So, a design defined by the setting of parameters η_1, η_2, and η_3 will in practice have a slightly different geometry, defined by a perturbed setting of these parameters. We assume that this effect can be simulated by drawing the implementation error from a normal distribution with a standard deviation as given in Table 1.

Since the relationship of Drop test and Picture quality expressed in terms of the design parameters are highly nonlinear, the approximating model type that was used is a Kriging model. First, we calculated the nominal solution. Then, we have applied the technique described in Section 6.4 to calculate a robust solution. To compare the resulting robust solution to the nominal one we have simulated a number of implementation error realizations. Given a realization of the implementation error, the maximum stresses in the nominal optimum and in the robust optimum are predicted. The results are depicted in Figure 6 and Figure 7.

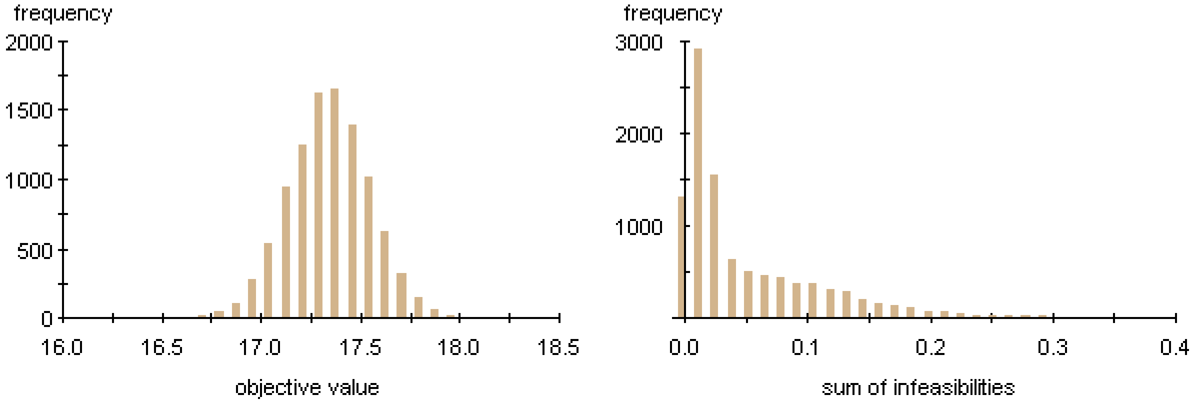

Figure 6. The objective values (left) and sum of constraint violations (right) of the nominal mask optimization problem in 10.000 samples.

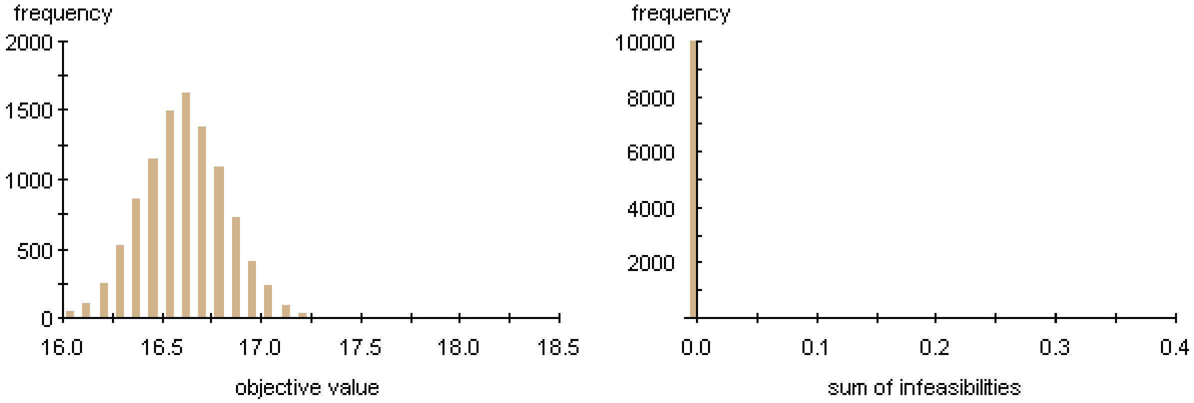

Figure 7. The objective values (left) and sum of constraint violations (right) of the nominal mask optimization problem in 10.000 samples.

It appeared that when the nominal optimal design is used, only 13.1% of the designs are feasible. When the robust optimal design is used, 99.7 % of the designs are feasible. Note however, that the robust solution leads to a reduction in the objective value.

8 Conclusions and further research

In this paper we have argued that there are three types of errors made when metamodels are used to optimize a design or process. The simulation-model error receives little attention in the literature, while in practice this error may have a significant impact. The robust counterpart methodology can be used to obtain robust solutions, i.e., solutions which are less sensitive with respect to the errors. For several metamodels (i.e. linear, quadratic, linear regression and Kriging models) and for different types of errors we have developed solvable robust counterpart optimization problems. For an overview, see Table 2. Finally, we have shown that these techniques can successfully be applied to find robust solutions for two TV tube design problems. The effect of the simulation error appeared to be significant, and the robust solutions found in this paper are much better than the nominal optimal solution.

In this paper, we treated the three errors (simulation error, metamodel error, and implementation error) separately. An interesting subject for further research is to analyze how all these errors can be modeled in one robust counterpart problem simultaneously.

Error type ↓		***Metamodel*** →	**Linear**	**Quadratic**	**Lin. Regression**	**Kriging**
Simulation-model error	**Box**		LP	QP	NLP	NLP
	Ellipsoid		SOCO	NLP	NLP	NLP
Meta model error			NLP	NLP	NLP	NLP
Implementation error	**Box**	**Multiplicative**	LP	QP	NLP	NLP
		Additive	LP	QP	NLP	NLP
	Ellipsoid	**Multiplicative**	SOCO	NLP	NLP	NLP
		Additive	LP	NLP	NLP	NLP

Table 2: Overview of optimization problem classes for all robust counterpart problems.

References

Barthelemy, J.-F.M. and Haftka, R.T. (1993), Approximation concepts for optimum structural design – a review, Structural Optimization 5, 129-144.

Bates, R.A., Gilliver, R., Hughes A., Shahin, T., Sivaloganathan, S., and Wynn, H.P. (1999), Fast optimization of mechanical designs using computer aided design computer aided engineering emulation: a case study, Proceedings of the Institution of Mechanical Engineers, Part D-Journal of Automobiel Engineering 213 (D1), 27-35.

Bates, R.A., and Wynn H.P. (1996), Tolerancing and optimization for model-based robust engineering design, Quality and Reliability Engineering International 12 (2), 119-127.

Ben-Tal A, and Nemirovski A. (2002) Robust optimization - methodology and applications, Mathematical Programming 92 (3), 453-480.

Doltsinis, I., Kang, Z. (2002), Robust design of structures using optimization methods, Computer Methods in Applied Mechanics and Engineering 193 (23-26), 2221-2237.

Du, X., Wang, Y., and Chen, W. (2000), Methods for robust multidisciplinary design, AIAA 2000-1785, 1-10.

Efron, B., and Tibshirani, R.J. (1993), An introduction to the bootstrap, Number 57 in Monographs on statistics and applied probability. Chapman & Hall, New York.

Gu, X., Renaud, J.E., Batill, S.M., Brach, R.M. and Budhiraja, A.S. (2000), Worst case propagated uncertainty of multidisciplinary systems in robust design optimization, Structural and Multidisciplinary Optimization 20 (3), 190-213.

Hertog, D. den, Klerk, E. de , and Roos, C. (2002), On Convex Quadratic Approximation, Statistica Neerlandica 563, 376-385.

Hertog, D., den and Stehouwer, H.P. (2002), Optimizing color picture tubes by high-cost nonlinear programming, European Journal on Operations Research 140, 2, 197-211.

Hertog, D. den, Kleijnen, J.P.C., and Siem, A.Y.D. (2004), The correct kriging variance estimated by bootstrapping, CentER Discussion Paper 2004-46, Tilburg University, Tilburg. To appear in Journal of the OR Society.

Jin, R., Du, X., Chen, W. (2003), The use of metamodeling techniques for optimization under uncertainty, Structural and Multidisciplinary Optimization (2), 99-116.

Jones, D., Schonlau, M., and Welch, W. (1998), Efficient global optimization of expensive black-box functions, Journal of Global Optimization 13, 455-492.

Kleijnen J.P.C., and van Beers, W.C.M. (2004), Application-driven sequential designs for simulation experiments: Kriging metamodelling, Journal of the Operational Research Society, 55 (8), 876-883.

Koch, P.N, Mavris, D., Allen, J.K., and Mistree, F. (1998), Modeling noise in approximation-based robust design: a comparison and critical discussion, in: Proceedings of DETC'98, 1998 ASME Design Engineering Technical Conferences, Atlanta, Georgia, 1-15.

Koch, P.N., Yang, R.J., Gu, L. (2004), Design for six sigma through robust optimization, Structural and Multidisciplinary Optimization 26 (3-4), 235-248.

Lophaven, S.N., Nielsen, H.B., and Sondergaard, J. (2002), Dace: A matlab kriging toolbox version 2.0. Technical report IMM-TR-2002-12, Technical University of Denmark, Copenhagen.

Mavris, D.N., Bandte, O., Schrage, D.P. (1996), Application of probabilistic methods for the determination of an economically robust HSCT configuration, AIAA-96-4090, 968-987.

Montgomery, D.C., Peck, E.A., and Vining, G.G. (2001), Introduction to linear regression analysis, 3rd ed., Wiley, New York.

Putko, M.M. Newman, P.A., Green, L.L. (2001), Approach for uncertainty propagation and robust design in CFD using sensitivity derivatives, AIAA 2001-2528, 1-14.

Rajagopal, R, and Del Castillo, E. (2005), Model-robust process optimization using Bayesian model averaging, to appear in Technometrics.

Sacks, J., W.J. Welch, T.J. Mitchel, and H.P. Wynn (1989), Design and analysis of computer experiments, Statistical Science 4, 409-435.

Sanchez, S.M. (2000), Robust design: seeking the best of all possible worlds, Proceedings of the 2000 Winter Simulation Conference, eds. J.A. Joines et al., 69-76.

Sobieszczanski-Sobieski J., and Haftka R.T. (1997), Multidisciplinary aerospace design optimization: survey of recent developments, Structural Optimization 14 (1), 1-23.

Stehouwer, H.P., and Hertog, D. den (1999), Simulation-Based Design Optimisation: Methodology and Applications (extended abstract), Proceedings of the first ASMO UK / ISSMO conference on Engineering Design Optimization, Ilkley, UK.

Sturm, J.F. (2002), Implementation of interior point methods for mixed semidefinite and second order cone optimization problems, Optimization Methods & Software 17 (6), 1105-1154.

Su, J., and Renaud, J.E. (1997) Automatic differentiation in robust optimization, AIAA 35, 1072-1079.

Waldron, S. (1998), Multipoint Taylor formulae, Numerische Mathematik 80 (3), 461-494.

Watson, G.A. (2004), Robust solutions to a general class of approximation problems, SIAM Journal on Scientific Computing 25, 1448-1460.

Xu, D., and Albin, S.L. (2003), Robust optimization of experimentally derived objective functions, IIE Transactions 35 (9), 793-802.