DESIGN OPTIMISATION: SOME PITFALLS AND THEIR REMEDIES

Erwin Stinstra, Lonneke Driessen, Peter Stehouwer

Centre for Quantitative Methods, P.O.Box 414, 5600 AK Eindhoven, The Netherlands

E-mail: DOT@cqm.nl

ABSTRACT

Designs are frequently optimised using Computer Aided Engineering (CAE) tools in combination with a specialised optimisation strategy. In this paper we describe a number of pitfalls that can be encountered when using such a strategy. Furthermore, we describe their remedies. Some remedies are demonstrated using a real-life case in picture tube design.

Keywords: Design optimisation, pitfalls, picture tube design.

1. INTRODUCTION

Nowadays, products are more and more often designed on the computer. Designers use Computer Aided Engineering (CAE) tools for virtual prototyping. Although much faster than real prototyping, CAE simulations are still very time-consuming. Due to freedom in design, there usually are astronomical numbers of alternatives. Furthermore, products have to meet many, possibly conflicting, design requirements originating from various engineering disciplines. To cope with this increasing complexity, techniques from statistics and mathematical optimisation, like Design of Experiments (DoE), Response Surface Modelling (RSM) (see Khuri et al, 1996), and Mathematical Programming, are more and more recognised as being indispensable in this area.

However, these techniques are easily misused. The aim of this paper is to describe a number of pitfalls that may lead to wrong or inefficient use of these techniques and wrong interpretation of the results. Examples of such pitfalls are inappropriate model validation, local optima, and non-robust optima. We present a methodology to avoid these pitfalls, which is illustrated with a real-life application. This application originates from colour picture tube design and concerns the optimisation of the geometry of an electron gun, which is a complex non-linear optimisation problem.

This paper is organised as follows. In Section 2 we describe our approach to design optimisation. Section 3 describes pitfalls that we encountered, possible negative effects, and remedies that can be used to prevent them. Finally, in Section 4 we describe a practical case in which we illustrate how to handle some of the pitfalls correctly. We conclude this paper with some remarks.

2. DESIGN OPTIMISATION METHODOLOGY

We developed an optimisation approach that elaborates and extends on both RSM and what is called Design and Analysis of Computer Experiments (DACE); see Myers (1999) and Sachs et al (1989).

The approach, implemented in the design optimisation tool COMPACT developed by CQM, consists of the following four steps:

- Problem specification
- Generation and simulation of an experimental design
- Compact modelling
- Prediction, optimisation and robust design

In the sequel, each of these steps will be described in short. For a more detailed description we refer to den Hertog and Stehouwer (1999).

Problem specification

In the first step of the design optimisation methodology a mathematical problem specification is set up. This is a very important and often quite difficult phase. In the subsequent phases design decisions are made, based on the models that are developed according to specifications set up in the problem specification phase.

In this phase, we have to decide which design parameters are most important, which response parameters are needed for judging the quality of a specific design and which objective function best captures the optimality requirements of a design. Furthermore, design restrictions have to be modelled and it has to be decided how many simulation runs are going to be performed.

Generation and simulation of an experimental design

After the specification of the problem, a simulation scheme is generated. This scheme consists of a set of simulation runs that are chosen such that they are located within the feasible design region. The design region is that part of the design parameter space that satisfies all bounds on design parameters defined during the problem specification phase. Once the experimental design has been created, the runs arising from this scheme are evaluated by means of simulation.

Compact modelling

The third step aims at obtaining a good model description in terms of design parameters for each response parameter. We call such models compact models as opposed to the detailed CAE-models. These models are based on the results of the simulations performed in the second step. The compact model types we use are first- and second-order polynomial models (Montgomery, 1984) and Kriging models (Sachs, et al, 1989). Generally speaking, the latter models yield the best approximations when the underlying relationship has a highly non-linear structure with multiple local optima. Statistical model selection criteria are used to assess how well a compact model fits the underlying relationship.

Prediction, optimisation and robust design

The first three steps result in a compact model for each of the response parameters. In the fourth step these compact models are used for prediction, optimisation and robust design. We explain these notions next:
Prediction: Using the compact models, the values of the response parameters can be predicted for any new design that is located in the feasible design region. Furthermore, possible infeasibilities due to response parameter bounds as well as the objective value of the new design are predicted.
Optimisation: Optimisation is a method to find feasible settings for the design parameters that minimise (or maximise) the objective function. The optimisation is performed by incorporating the compact models in powerful linear programming (LP) and non-linear programming (NLP) solvers.
Robust design: Monte Carlo techniques are used to analyse the robustness of a certain design to random deviations in the design parameters. Note that as a direct result of fast prediction with help of the compact models, Monte Carlo simulations can now be carried out quickly.

3. PITFALLS AND THEIR REMEDIES

In this section we describe for each phase of our design optimisation approach a number of important pitfalls that we encountered in practice. For each pitfall we indicate the possible negative effects on the design process if it is not handled correctly. After that, we describe a remedy.

Problem specification

The importance of choosing suitable design and response parameters should not be underestimated. Often it is not obvious which design parameters must be taken into account. A too complex parameterisation should be avoided, but at the same time it must be complex enough to represent the real-life design in a satisfying way.

Fitting compound responses
Compound responses are known functions of the response parameters, that arise in the objective or the constraints of the optimisation problem. For example, it is often necessary to minimise the maximum over a number of responses. In the gun optimisation case described in Section 4, we give an example of such a compound response. There are two approaches to handle such compound functions:

- Generate one compact model for this compound response.
- Generate compact models for the original (non-compound) responses, and substitute them into the known compound function during prediction and optimisation.

We observe that often the first approach is followed, while in our opinion the last choice is the best one. The main reason is that in the second approach the known function structure is used, whereas in the first approach this has to be re-discovered by the compact model. As a result, the compact model for the compound response is usually less accurate than the models fitted to the individual responses. This has a negative effect on the prediction power of the models.
Figure 1 illustrates both approaches. The x-axes represents a design parameter, the y-axes represents a response value. Fitting compact models for the individual responses depicted in Figure 1a is much easier than fitting a compact model for the compound response shown in Figure 1b. Note that following the approach of fitting individual responses results in more response models. This is usually not a problem though, since the amount of time needed to fit a compact model is usually negligible when compared to a simulation run.

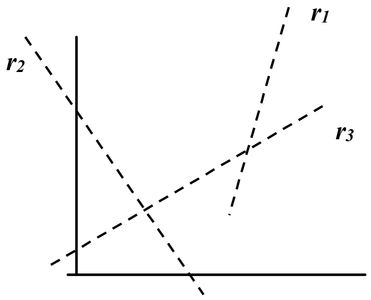

Figure 1 a. Three non-compound responses

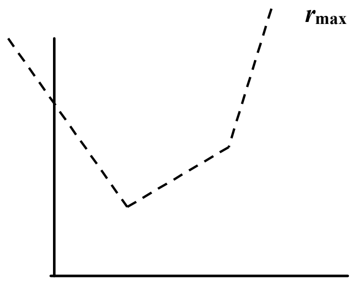

Figure 1 b. One compound response

Parameterisation issues
Another important issue is the question what design parameters to choose and how to present them in the design optimisation problem. The importance of transformations of design or response parameters is often underestimated. This can lead to unnecessary highly non-linear compact models that do not give the designer much insight, whereas linear models might have been sufficient after a suitable transformation.
Another situation that should be prevented, is a redundant parameterisation. For example, symmetry may cause multiple different runs to specify the same design. In such a parameterisation, usually a too large design space is modelled. Of course, using different design parameters or restrictions on combinations of design parameters can prevent this pitfall.
The last situation we want to stress here, is the case in which there are integer-valued design parameters involved. There are many design parameters for which non-integer values do not make sense, for example, the number of holes in a board. It can also be the case that a certain product can only be obtained in a few pre-specified measures. Often these integer parameters are treated as continuous ones during the optimisation process. Our experience is that this solution is a good one as long as the integer design parameter is 'almost continuous', i.e., the range of this design parameter is still so large that round-off errors are not expected to have big impact on the quality of the final optimal design. But when there are only a few possible values for a certain design parameter, rounding off this design parameter might lead to a sub-optimal design.
Another pitfall concerning integer design parameters is the use of an integer quantitative parameter to model a qualitative aspect of the design. For example, one may use an integer code to indicate a colour and subsequently use the code to create a compact model. Of course this leads to useless compact models. We go deeper into this subject when we discuss pitfalls in the optimisation step.

Use of too many design parameters
Another pitfall is the use of too many design parameters. This leads to what is often called the 'curse of dimensionality': the number of required simulation runs increases rapidly as the number of design parameters increases. Furthermore, the construction of a good simulation scheme and the optimisation become very time-consuming. This pitfall can be prevented by using screening to select the most important design parameters or by using a different optimisation approach that is specially suited for high-dimensional problems. In such cases, we use an iterative method. This method uses all information of previous simulations to generate a promising new simulation run. When enough simulation results are available, local approximating models of the simulation tool are created. The approximating models are then locally optimised within a so-called trust region to find the best feasible objective improving run. This trust region moves along the most promising direction. In every iteration, a new local linear approximation is built, and either a new simulation is evaluated or the trust region is decreased. For examples and detailed descriptions we refer to the work of Conn & Toint (1996), Powell (1994), and Toropov (1992).

<u>Generation and simulation of an experimental design</u>

The problem of choosing a simulation scheme is called *Design of Experiments* (Montgomery, 1984). There is a vast range of experimental designs that all have their advantages and disadvantages. In the sequel, we describe a number of pitfalls in creating an experimental design.

Classical DoE is used for deterministic situations
Classical DoE has been developed for and is therefore mainly focused on physical experimentation in which experiments and measurements of response parameters are subject to noise. As a result, these schemes have certain characteristics that, although advisable for experimentation under uncertainty, are undesirable in deterministic situations. In computer experimentation noise usually does not play a role, since running a computer simulation twice generally yields exactly the same results. Therefore, no additional information is gained from the repeated simulation of the same simulation run, which is often proposed in classical DoE.
Also due to the presence of noise, in physical experimentation it is often optimal to have design points that lie on the borders of the design region. In computer

experimentation, the inner part of the design region is equally interesting.

Summarising, a computer simulation scheme generated by classical DoE may contain replications, and may lead to simulation runs that are not evenly divided over the design region. The first characteristic results in waste of time, the second in bad approximating models. In our opinion simulation schemes for computer experimentation must be *space filling* and *non-collapsing*. In a space-filling design the evaluation points are spread out over the feasible design region as evenly as possible. A design is non-collapsing if no two simulation runs have the same value for any design parameter. An example of a non-collapsing simulation scheme is a Latin Hypercube Design (LHD).

Using a space-filling and non-collapsing scheme, none of the time consuming simulations may become useless. We therefore propagate to use a space-filling Latin Hypercube Design for computer experiments, specially when no information about the response parameters is available.

Design constraints are neglected which results in a larger design region than necessary

From our design optimisation practice we experienced that the feasible design region is often a non-box region. For example, this happens when simulation runs in some part of the box constrained region have no physical interpretation or cannot be simulated. Moreover, it is always better to use prior knowledge on uninteresting or infeasible parts of the design space when creating a simulation scheme. The larger the design region, the more simulation runs are usually needed to be able to fit accurate approximating models. Therefore it is desirable to restrict the design region as much as possible by adding a priori known restrictions on combinations of design parameters. A drawback of most classical DoE methods is that they are only applicable for rectangular design regions. Use of such a classical method almost automatically forces the designer to work with a box-shaped design region. To prevent this, we created a methodology to construct a constrained space-filling LHD (Stehouwer, Stinstra, Vestjens, to appear). This method can deal with any type of additional constraints on the design parameters, as illustrated in Figure 2.

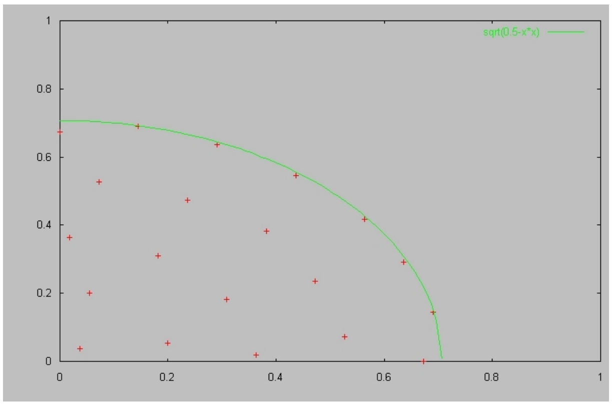

Figure 2. A two-dimensional constrained Space-filling LHD

Use of a too large simulation scheme at once

Although every designer tries to develop a new product or process within as few simulation runs as possible, they often still perform all the simulations at once. This can lead to unnecessary simulation runs. It is better to first perform a small set of simulations, fit linear models and check whether these models are of sufficient quality. If they are not, some additional simulations can be performed and with help of the newly gathered information more complex models can be fitted. Of course, the second set of simulations should be chosen such that together with the first set they cover the design region as well as possible. Our methodology for the construction of a constrained space-filling LHD can generate new simulation runs given an already existing set of runs. The resulting design is non-collapsing and space-filling.

Model generation

Given the simulation results, it is important to create the best models possible. The pitfalls that we encountered in this phase can lead to less insight and incorrect optimisation results.

Unnecessary complex modelling

One of the pitfalls that we encountered in this step is the use of too complex models. It is often assumed that a more complex model should be used in any case, even if a simpler model is enough. This is dangerous for the following reasons:

- A more complex model represents the data, while it should represent the trends. A model with enough degrees of freedom can perfectly fit through any data set. But these models generally do not predict very well.
- The more complex the models are, the more complex the optimisation will become. This often results in the need for global optimisation strategies when the problem becomes non-convex.
- More complex models like interpolators usually give the designer less insight into the underlying relationships.

Simpler models should thus be preferred. A good way-of-working is to start with a linear model. If the model is not good enough, a quadratic model should be created. Note that this does not necessarily need to be a full quadratic model, i.e., a model with all interactions and quadratic terms. When a linear model does not fit well enough, we use a *pruning* strategy. This works as follows. First a full quadratic model is fit. Next, the least important term is removed from the model. The model is then rebuilt. If the new model performs better than the old, the removed term is deleted permanently and the least important term in the new model is removed. We continue this procedure until no improvement is found.
If the pruned quadratic model is not good enough, we use an interpolating Kriging model (Sachs et al,1989).

Lack of model validation
Model validation is very important. Compact models are usually validated by calculating some validation statistic or cost function of the differences between simulated and predicted values. An example of such a statistic is the mean squared error function. Usually, such validation statistics are calculated on the same data set as was used to build the compact model. The lower the mean squared error, the better the model mimics the data. We call this approach the *naïve validation approach*. This leads to the disadvantage that it does not account for the effect of over-fitting: by making the compact model arbitrary complex, the function value can be made arbitrary small. In the extreme case of interpolating compact models (e.g., Kriging models, splines or sufficiently high-ordered polynomials) the value of such a cost function can easily be made equal to zero.

It is not the mimic capabilities but the prediction capabilities of a model that we want to assess. We propagate the use of one of the following techniques for validating the prediction capabilities of a compact model.

- Independent test set – Assess the prediction capabilities of the compact model on an independent test set and calculate the desired validation statistics. A major disadvantage is that extra, often time-consuming, simulations have to be performed.
- Cross-validation – Re-estimate the compact model *n* times, with *n* equal to the number of data points, while each time skipping one of the data points. Every time the skipped data point is used to test the prediction capabilities by calculating the desired statistic.

Optimisation

In the last step of our design optimisation approach, we encountered four possible pitfalls.

Naïve optimisation approach
Designers often optimise their design one factor at a time. This means that they first try to find the optimal setting for the first design parameter, then fix this parameter, and continue this procedure for the other parameters until all design parameters are optimised given the fixed values of the already optimised design parameters. This approach can lead to an ‘optimal’ design that is not even locally optimal when non-linear compact models are used or constraints on combinations of design parameters are present.
A much better approach is to use optimisation techniques from mathematical programming. The compact models can be incorporated in powerful LP or NLP solvers, like CONOPT (Drud, 1994).

Multiple local optima
When the NLP problem is non-convex, which is generally the case when interpolating models are used, the solution found by the solver will not automatically be the global optimum. This can lead to an incorrect ‘optimal’ design, which is locally optimal but not globally.
In such cases, it is wise to use a global optimisation strategy. We propagate the use of a multi-start optimisation technique. We start the local solver in several points in the feasible region and we select the best solution the solver returns. Of course, the starting points should be chosen in a clever way. In Step 2 of the compact model approach a space-filling Latin Hypercube Design was generated as a simulation scheme. We use these points as starting points for the local solver. Since they are well spread over the feasible region, we more likely end up with a global optimum.

Robustness issues
Once the optimal design has been found, it is important to consider its robustness. A pitfall that we frequently encounter, is the ignoring of the robustness of a certain design to random deviations in the design parameters. This can lead to unstable designs that are very sensitive to production tolerances.
To prevent this instability, sometimes Monte Carlo techniques are used directly with CAE tools, which is usually quite time-consuming. However, using compact models, Monte Carlo sampling becomes very fast and enables the designer to analyze the robustness of a certain design to random perturbations.

Integer optimisation
When some of the design parameters must have an integer value, this is commonly achieved by rounding off the optimal design parameter values to the nearest integer value. This rounding of can lead to completely wrong optimal settings, as can be seen in the two dimensional design space in Figure 3. The dots indicate the feasible integer designs.

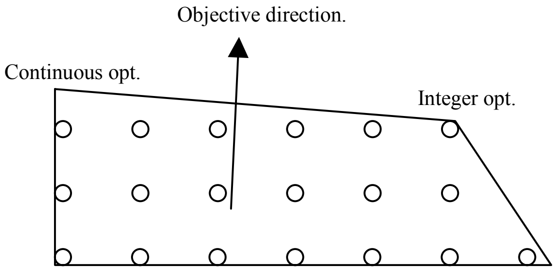

Figure 3. The continuous optimum and the integer optimum

This can be prevented by recognising the integer parameters as integer variables in the mathematical program and by consequently using a special mixed integer (non-) linear programming solver. For an overview of methods, see Biegler et al (1997).

4. OPTIMAL GUN DESIGN

During several projects, we have optimised several parts of the colour televisions at Philips. One of the optimised parts is the electron gun, which may be regarded as the heart of the colour picture tube. It generates the three electron beams, accelerates them, and ensures that they are focussed on the screen. Figure 4 shows a picture of an electron gun, containing a series of grids with holes.

Figure 4. Electron gun

The action of the electron gun is determined by the geometry of these grids and the application of different electrical voltages to them. The essence of gun design is to find the grid geometry that results in high quality guns. Among the most important quality characteristics are the *spot sizes* at several screen positions and at several applied voltages.

The geometry of the grid is parameterised by seven parameters; see Table 1.

Design parameter	Physical meaning
t_2	The thickness of the second grid
x_2	The horizontal size of the 2nd grid
y_2	The vertical size of the 2nd grid
s_{23}	The distance between the 2nd and 3rd grids
x_4	The radius of the circle-shaped 4th grid
x_5	The horizontal size of the 5th grid
y_5	The vertical size of the 5th grid

Table 1. The design parameters

Figure 5 schematically depicts the definitions of the design parameters. All design parameters are bounded by a lower bound and an upper bound.

The quality of a design is defined by the following response parameters:
- The spot sizes in the centre of the screen
- The spot sizes in the north-east corner of the screen

Due to symmetry, the spot sizes on the entire screen are modelled by these response parameters. The spot sizes are measured in x- and y direction and at 6 currents. This results in a total of 24 response parameters.

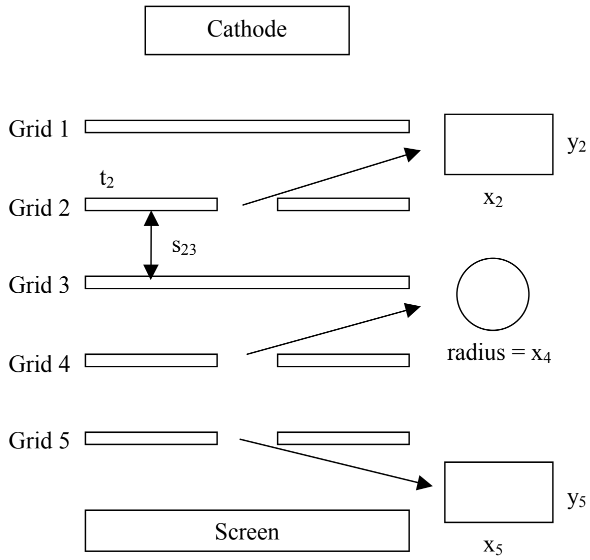

Figure 5. The grid structure of the gun

The design must be optimised with respect to the maximum spot size, i.e., the maximum spot size of the three electron beams on both positions, in both directions and using any current must be minimised.

A naïve formulation of the problem is the following:

$$\min \max_r (f_r(d_1,\ldots,d_7))$$
$$s.t.\ lb_i \le d_i \le ub_i, \qquad i = 1,\ldots,7$$

where $f_r(d_1,\ldots,d_7)$ denotes the value of compact model r in the design scenario $(d_1,\ldots,d_7)$. A better formulation of the optimisation problem is reached by breaking down the compound response (see Section 3) and introducing a dummy variable z that will converge to the maximal spot size. The reformulated problem looks as follows:

$$\min z$$
$$s.t.\ z \geq f_r(d_1,\ldots,d_7), \qquad \forall r \in R,$$
$$lb_i \leq d_i \leq ub_i, \qquad i = 1,\ldots,7$$

For this study, a simulation budget of 100 simulation runs was available. Using a spacefilling Latin Hypercube Design, the 100 simulations were optimally spaced through the feasible design space. Four simulations failed due to numerical reasons. Of the other 96 simulation runs, the best one has a maximum spot size of 6.59.
Next, we created polynomial models. Using cross validation, we concluded that the behaviour of nearly all response parameters is so non-linear, that polynomial models of second degree are not accurate enough. Therefore we used interpolating Kriging models in this case. The cross-validation statistic on these models is significantly better. For example, see the model validation statistics in Table 2 for a comparison of a quadratic model with the Kriging model for the spot size at 5 mA in the centre in y-direction. Note that the R^2, adjusted R^2 and RMSE give no information about the interpolating model.

	Quadratic model	Kriging model
R^2	0.41	1.0
Adj. R^2	0.23	1.0
RMSE	0.21	0.0
Cross Val.	0.29	0.25

Table 2. Model validation statistics

For optimisation, we used the global optimisation approach described in Section 3. Using this technique, we found several local optima, whose objective values are given in Figure 6. The best local optimum that we found has a predicted objective value of 5.25.

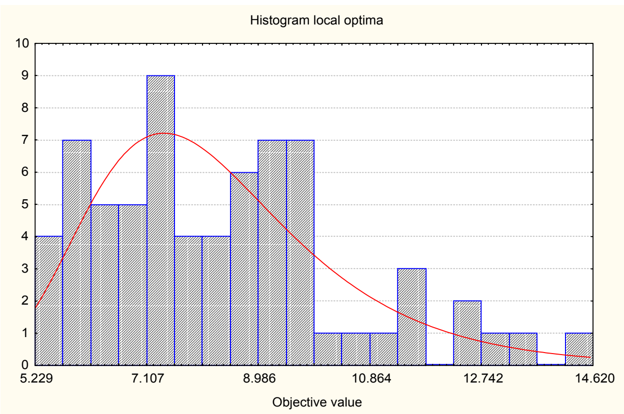

Figure 6. Histogram of local optima

The predicted optimum is significantly better than the best simulation. Of course, the predicted optimal scenario is simulated before the design is actually used. The predicted spot sizes in the optimum are depicted in Figure 7. Note that the maximum of 5.25 is obtained at several locations. This behaviour is typical when minimising a maximum.

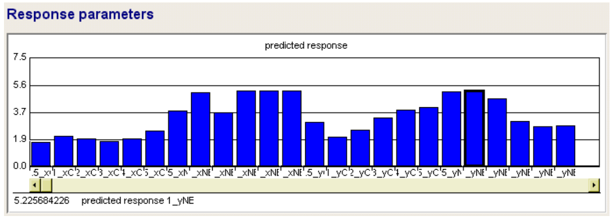

Figure 7. The optimal response values

Due to tolerances, for example in the production process, finding the optimal setting is usually not enough. We have to look into the effect of the instability of design parameters in the objective value. To show this, we assume a normal distribution on the design parameters. Using the response surface models, a histogram can be created using Monte Carlo simulations. This procedure, using 20,000 samples, results in the histogram depicted in Figure 8. Note that small disturbances in a locally optimal design always lead to a deteriorated objective value.

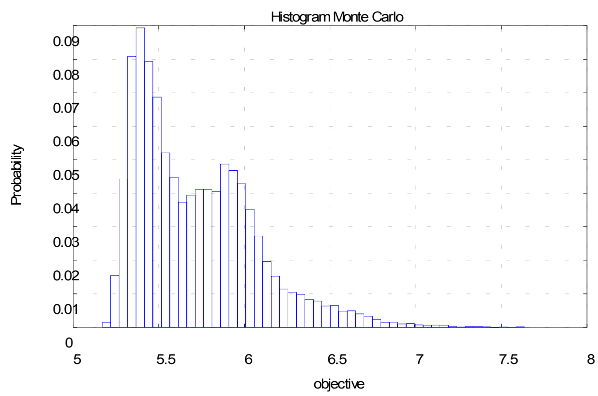

Figure 8. Histogram of the optimal design

We conclude that the local optimum that we found is sensitive for variability in the design parameters. We therefore start looking the second, third and fourth best local optima.

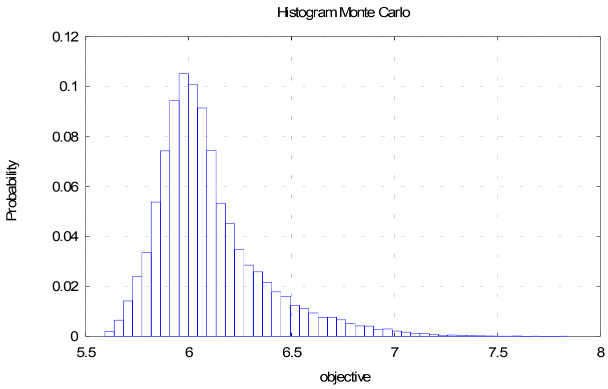

Figure 9. Histogram of the 4th best local optimum

The histogram of the fourth best local optimum is depicted in Figure 9. The nominal and expected value are a bit higher, the variance of the maximal spot size is somewhat lower.

CONCLUSION

Products are more and more often designed on the computer. This has led to extensive research in design optimisation. A number of general-purpose design optimisation tools is nowadays available. Decisions that need to be made when using such a tool are often not trivial. It is therefore in our opinion very important that designers gain insight in what happens when applying these techniques. What can go wrong can easily be illustrated by the pitfalls in this paper.

Besides helping designers to use design optimisation, the design optimisation community has the responsibility to create new tools and techniques that make it easier to design in a more and more complex world. Our current research is focussed at incorporating robust design in the optimisation step, i.e., minimise not only the expected objective value, but also the variance due to design instability. This functionality will be implemented in COMPACT. Furthermore, we are currently looking at integrating a mixed integer non-linear programming solver in our software in order to tackle the optimisation problems that integer design parameters introduce. Next to the compact model approach, we are working on the sequential toolbox SEQUEM, which enables us to solve cases that are large in terms of design parameters.

ACKNOWLEDGEMENT

This work was funded by Philips Components, Business Group Display Components, Product & Process Development.

REFERENCES

Biegler, L., Coleman, T., Conn, A., Santosa, F., (1997), Large-Scale Optimization with Applications, Part II: Optimal design and control, in *The IMA Volumes in Mathematics and its Applications*, Volume 93, Springer-Verlag, 73-100.

Conn A.R. and Ph.L.Toint (1996), An algorithm using quadratic interpolation for unconstrained derivative free optimization, in: *Nonlinear Optimization and Applications*, G. di Pillo and F. Giannes (eds.), Plenum Publishing, 27-47.

Drud, A.S. (1994), *CONOPT – A Large Scale GRG Code*, ORSA Journal on Computing 6, p. 207-216.

Hertog, D., den and Stehouwer, H.P. (1999), High Cost Nonlinear Programming, *European Journal on Operations Research*, submitted.

Khuri, A.I., Cornell, J.A. (1996), *Response surfaces: designs and analyses*, 2nd edition, M. Dekker (ed.), New York.

Morris, M.D. and T.J. Mitchell (1995), Exploratory designs for computer experiments, *Journal of Statistical Planning and Inference* **43**, 381-402.

Montgomery D.C. (1984), *Design and Analysis of Computer Experiments (Second Edition)*, John Wiley & Sons, New York.

Myers, R.H. (1999), Response surface methodology – current status and future directions, *Journal of Quality Technology* **31**, 30-74.

Powell, M.J.D. (1994), A direct search optimization method that models the objective and constraint functions by linear interpolation, *Advances in optimization and numerical analyses*, Proceedings of the sixth workshop on optimization and numerical analyses, Oaxaca, Mexico, Vol. 275, Kluwer Academic Publishers, Dordrecht, 51-67.

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

Stehouwer, H.P. ,Vestjens, A.P.A., Stinstra, E.D., *Constrained spacefilling Latin Hypercube Designs for computer experiments*, to appear.

Torczon, V., Trosset, M.W. (1998), Using approximations to accelerate engineering design optimization, *7th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization: A Collection of Technical Papers*, Part 2, 738-748.

Toropov, V.V. (1992), Multipoint approximation method in optimization problems with expensive function values, in: *Computational System Analysis*, Sydow, A. (ed.), Elsevier, 207-212.