THE ARCHIVES OF TRANSPORT

VOL. XXII NO 1 2010

10.2478/v10174-010-0008-z

Embedded Direct Search of Optimal Designs for Finite Noise Experiments

Franziska Schulz*
Kurt Frischmuth**

Received November 2009

Abstract

We study experimental designs for the identification of nonlinear model parameters. As optimality criterion we assume minimality of the error in a huge number of identifications run on simulated data, which are generated with known parameters and a given error distribution. The optimal design depends on the nonlinear parameters. We find the optimal solution set by combining a path following strategy and a direct search method.

Keywords: parameter identification, nonlinear regression, embedding method, direct search algorithm

1. Introduction

In many applications model parameters have to be identified from measurements or experiments. Some- times the number of observations is restricted by cost or time requirements. One way to avoid such restrictions is to replace real-world experiments by computer simulations, cf. [10] and cited there papers. The classical approach, however, consists in minimizing – in an appropriate sense – the deviation of the results by choosing details of the measurements in an optimal way. This may mean the determination of an optimal time for weighing an animal or an optimal spot for placing a sensor. Generally, we speak about an optimal experimental design.

In this paper, we study first a nonlinear model in one independent variable, which can be identified with time. There are three model parameters, one of them essentially nonlinear. This model serves mainly to demonstrate the concepts of our

* Universitat Rostock
** Universitat Rostock

approach. Next, we introduce a model in two input variables, which additionally depends on one further linear and one more nonlinear parameter.

The model is defined by an equation of the form

$$y = f(x; p) + e \tag{1}$$

Here we denote by e some random error, characteristic for the experiment we desire to optimize. It is most frequently assumed that for each single experiment the error is normally distributed, and that the errors are independent. Of course, the measurements should not be biased, i.e. the expected value of e should be zero.

For a given vector of parameters $p \in R^q$, $q \in N$, and a known distribution of the random error, we may generate simulated data, and then calculate for each sample the parameter p which fits best in the sense of the least sum of squares. Under suitable regularity assumptions, we obtain asymptotically a normal distribution of p around its true value – which was used for generating the data. Unfortunately, for realistic values of the errors, i.e. for finite noise, calculated solutions to the nonlinear regression model are far from a q-dimensional normal distribution. The results fill rather a banana-shaped domain in parameter space. Consequently, instead of minimizing e.g. the volume of confidence regions (classical D-criterion) or projections onto chosen axes in parameter space, cf. [1], we propose to define optimality in terms of the cloud of results obtained by simulation.

A well known problem in experimental design for nonlinear problems is the dependence of the optimal design on the nonlinear parameters. In fact, before we perform the first measurements, the parameters are not at our disposal, at best we may have some rough initial guess. One way out of this dilemma is a sequential approach, which consists in estimating the nonlinear parameters anew after each measurement, and adjusting the design according to the new information. Of course, if the design variable is time, this may lead to problems this causality.

Hence we need not only a single optimum design, but rather a mapping from the sub-space of nonlinear parameters to the corresponding optimal designs, by which they could be identified best.

2. Nonlinear Regression Model

We start with a well known model of the assumed form

$$y = f(x; p) + e = a + becx + e \tag{2}$$

The unknown parameters a, b and c constitute the q = 3 – dimensional parameter vector p. The independent variable x is restricted to an observation interval $[x_l, x_r]$. We assume that we know the distribution of e so that we can generate the error terms.

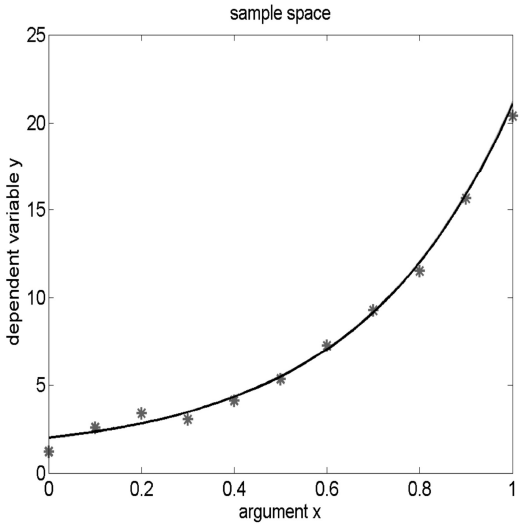

Fig. 1. Nonlinear regression model

We chose arbitrarily the true parameter vector to be $p_0 = (100, -70, -3)^T$ and $[x_l, x_r] = [0, 1]$. For a given sample of observations $y = (y_1, y_2 \ldots y_n)^T$ at measuring points $x = (x_1, x_2 \ldots x_n)^T$ we calculate the least squares estimate by minimizing

$$F(p; x; y) := \frac{1}{2} \sum_{j=1}^{n} (y_j - f(x_j; p))^2 \tag{3}$$

with respect to the unknown vector p. For our academic model, this problem is easy to solve numerically as far as we have a reasonably scattered set of measuring points x and the errors e are not excessive. The model we explore is nonlinear, as are the real-world problems we plan to solve in the future. Hence we need an initial guess for the unknown p. The obvious choice is to start with the known parameter vector p_0 we used for the simulations. There are several effective methods for solving the minimization problem given in (3). We base our procedure on the Gauss-Newton technique. For moderate (but finite, i.e. not infinitesimal) errors, we substitute the true model by the first order Taylor approximation, and solve the overdetermined linear system of equations for the improvement Δ p

$$y_j = f(x_j; p) + \nabla_p f(x_j; p)\Delta p \tag{4}$$

The most simple version of the algorithm, i.e. the iteration

$$p := p + \Delta p \tag{5}$$

converges for (really) good initial guesses, in general, however, certain improvements are recommended, such as damping, linesearch or some kind of regularization, e.g. as in Marquard and Levenberg's method. For details we refer to [5], but notice that the matrix

$$L(x, p) = \nabla_p f(x; p) \tag{6}$$

plays a crucial role as well in the analysis of the problem as in the numerical solution. In a linear model, L does not depend on p, and the solution can be simply written as

$$p(x) = (L(x)^T L(x))^{-1} L(x)^T y \tag{7}$$

This makes it obvious that large eigenvalues of $(\mathrm{L(x)}^T\mathrm{L(x)})^{-1}$ are not desirable, likewise small singular values of L(x) are bad, i.e. the columns of L(x) should be well linearly independent. This is an important criterion for the choice of the experimental design x.

Our problem, however, is nonlinear, and hence L alone does not contain the whole information about the quality of an estimate p calculated on the basis of measurements that are fairly hard affected by random errors. Instead of presenting an analysis of second order (and higher) terms, we use here a brute force approach.

We perform a high number of simulations, i.e., we add random errors to the exact values of y as calculated for a given vector of x-values and the fixed parameter p_0. Next, for each simulated y vector, we calculate p. The resulting random cloud is presented in Fig. 11.

3. Numerical Recepies

It is obvious that the presented approach relies on a large number of single calculations, and the results are the better, the more cases we are able to handle. Hence numerical efficiency is essential, and we put some efforts into the speed-up of the implementation of the endeavor. We singled out three crucial points for the overall performance, and devoted one subsection to each of them.

3.1. Massive Solutions of the Regression Problem

We perform n observations at the points $\mathrm{x}_1, \ldots, \mathrm{x}_n$ and repeat the measurements m times. So we obtain a matrix $\mathrm{Y} \in \mathrm{R}^{n\times m}$ with $\mathrm{Y}_{ij} := \mathrm{y}_i^{(j)}$, where $\mathrm{y}_i^{(j)}$ is the measured value of the jth repetition at the point x_i for i = 1, . . . , n and j = 1, . . . , m. We search for the solution $\mathrm{P} = (\mathrm{p}^{(1)}, \ldots, \mathrm{p}^{(m)}) \in R^{3\times m}$ of the least squares problem

$$\frac{1}{2}\sum_{i=1}^{n} (\mathrm{y}_i^{(j)} - f(x_j; p^{(j)}))^2 \to \min \forall j = 1, ..., m$$

For a faster implementation in Matlab, we want to avoid solving the least squares problem for each j = 1, . . . , m in a loop, and use the following matrix approach. For the exact parameters p = (a, b, c)^T we first define a matrix

$$M := \begin{pmatrix} \left(\nabla_p f(x_1, p)\right)^T \\ \vdots \\ \left(\nabla_p f(x_n, p)\right)^T \end{pmatrix} = \begin{pmatrix} 1 & e^{cx_1} & bx_1 e^{cx_1} \\ \vdots & \vdots & \vdots \\ 1 & e^{cx_{n1}} & bx_n e^{cx_n} \end{pmatrix} \in R^{nx3}$$

Then the Gauss-Newton method has the form

$$p_{k+1} := p_k - \lambda_k \circ (M^T M - \mu I)^{-1} M^T \begin{pmatrix} y_1^{(1)} - f(x_1, p_k^{(1)}) & \dots & y_1^{(m)} - f(x_1, p_k^{(m)}) \\ \vdots & \ddots & \vdots \\ y_n^{(1)} - f(x_n, p_k^{(1)}) & \cdots & y_n^{(1)} - f(x_n, p_k^{(m)}) \end{pmatrix}$$

for k = 0, . . . , k_{max} with

$$p_0 := p(1, \ldots, 1)^T = (p, p, \ldots, p) \in R^{3\times m}$$

and

$$\lambda_k = \begin{pmatrix} \lambda_k^{(1)} & \cdots & \lambda_k^{(m)} \\ \lambda_k^{(1)} & \cdots & \lambda_k^{(m)} \\ \lambda_k^{(1)} & \cdots & \lambda_k^{(m)} \end{pmatrix}$$

where $\lambda_k^{(j)}$ denotes the step size for optimizing p$^{(j)}$, μ a constant regularization parameter and $^\circ$ is the elementwise multiplication. After having computed p_{k+1} we compare the new residuum for each $y^{(j)^k}$ with the last one and divide the corresponding $\lambda_k^{(j)}$ by four if the residuum has increased. We repeat this procedure five times before we start a new iteration step.

Finally we reject the 1% of the parameter sets p$^{(j)}$, j = 1, . . . , m, that has the largest residuum. Due to clearness we denote the remaining parameter sets by p$^{(1)}$, . . . , p$^{(M)}$ with M = 0.01 m. Our objective function g : $R^n \to R$: (x) 7→ g(x) now is

$$g(x) := \max_{j = 1, ..., M} \|p - p^{(j)}\|_2 \tag{8}$$

So we measure our nonlinear model at the points x = (x$_1$, . . . , x$_n$)^T and the function value is the diameter of the resulting parameter cloud.

3.2. Direct Search of a Single Optimal Design

We choose the Nelder-Mead simplex method for optimizing the objective function in (8) (see [7]). This direct search method minimizes a nonlinear function $g : R^n \to R$ without using any derivatives. At each iteration step, the function values at the vertices of an n-simplex, where each of them represents a design, are compared with function values computed at one or more test points. The intention is to replace the worst vertex of the actual simplex by its reflection into a better direction.

Each iteration starts with sorting the vertices $x^{(1)}, \ldots, x^{(n+1)}$ by their values, i.e.

$$g(x^{(1)}) \geq g(x^{(2)}) \geq \ldots \geq g(x^{(n)}) \geq g(x^{(n+1)}).$$

We define the center of the n best points as $z := \frac{1}{n} \sum_{i=2}^{n+1} x^{(i)}$ and the reflected point as $r := z + \alpha(z - x^{(1)})$
with $\alpha > 0$. If $g(x^{(n+1)}) \leq g(r) < g(x^{(2)})$ then we obtained a relatively good test point, replace $x^{(1)}$ by r and start a new iteration with the simplex consisting of the vertices $r, x^{(2)}, x^{(3)}, \ldots, x^{(n+1)}$ (reflection step).

Otherwise, if $g(r) < g(x_{n+1})$, we compute a second test point $e_x := z + \beta(z - x^{(1)})$ with $\beta > \alpha$. If the function value of e_x is better than that of r, the worst point $x^{(1)}$ is replaced by e, and the current iteration step is complete. Otherwise we accept r (expansion step).

Instead of enlarging the distance between z and the test point, we reduce it if $g(r) \geq g(x^{(2)})$ and distinguish two cases: In the first one, we perform an outside contraction and compute $k^+ := z + \gamma(z - x^{(1)})$ with $\gamma < \alpha$ if $g(x^{(2)}) \leq g(r) < g(x^{(1)})$. The point $x^{(1)}$ is replaced by k^+ if it is better than r. Else we shrink the simplex and get a new simplex with the vertices

$$\frac{1}{2}\left(x^{(1)} + x^{(n+1)}\right), \frac{1}{2}\left(x^{(2)} + x^{(n+1)}\right), \ldots, \frac{1}{2}\left(x^{(n)} + x^{(n+1)}\right), x^{(n+1)} \tag{9}$$

We have to do an inside contraction in the second case if r is even worse than $x^{(1)}$. It means that we replace x_1 by the point $k^- := z - \gamma(z - x^{(1)})$ if it yields a better point than r. Otherwise we start a new iteration with the shrunken simplex defined by (9).

Using the Nelder-Mead simplex method for function minimization, we only need one function evaluation per iteration in the case of a reflection step. If we have to perform an expansion step or if we accept either k^+ or k^- two function evaluations are needed. A shrink step will costs n + 2 evaluations because we have to compute r, k^+ or k^- and the new n points in (9). Most of the evaluations are required in a shrink step, but Langarias et. al. showed in [8] that such a step never happens for strictly convex functions. They also proved some convergence properties of their method. Therefore it can be shown that the n + 1 sequences of the function values

at the simplex vertices contain a monotonic decreasing subsequence if the number of shrink steps is finite. Consequently, these sequences converge if the function g is strictly convex and bounded from below. This does neither mean that the n + 1 sequences have the same limit and the simplex vertices converge to the same point, nor that the result is a minimizer of g.

If, additionally, we assume that n = 1 and g has bounded level sets, convergence to a minimizer can be proved. Also in the case n = 2 the three sequences of the function values have the same limit and the simplex diameter converges to zero. In general, however, convergence results for higher dimensions than n = 2 are relatively weak up to now. Nonetheless, for practical purposes, the algorithm and its clones perform very well for most real-life cases for which gradient-based algorithms are not suitable and evolution type methods too expensive.

3.3. Systematic Search over the Parameter Space

Now we let the true parameter c in the nonlinear model (1), here f (p, x) = a + be^{cx}, vary in a suitable range. The objective function depends now on one additional parameter and takes the form

$$h : R^n \times [C_0, C_1] \to R : (x, c)7 \to h(x, c) \tag{10}$$

The optimizer x^* for a given parameter C_0 is supposed to be known and we are interested in the minimizer x of h(x, c) for each parameter $c \in [C_0, C_1]$.

For that we discretise the interval $[C_0, C_1]$ in grid points $C_0 = c_0 \leq c_1 \leq \ldots \leq c_N = C_1$. For each supporting point c_k, k = 1, . . . , N, we first compute a starting point using one or more optimizers for the previous grid points $c_0, \ldots, c_{k-1}$ and then use the Nelder-Mead simplex method with this starting point to obtain the optimum for c_k.

These homotopy optimization methods are predictor corrector methods. So we try to follow the path of the minimizers. Under appropriate assumptions the local existence and uniqueness of a solution curve result from the implicit function theorem, [9]. We also get a differential equation for the solution path depending on the homotopy parameter c out of it. If the necessary derivatives are available, we can use methods for ordinary differential equations as a predictor. Otherwise there are extrapolation predictors. The simplest possibilities are using the latest computed minimizer, or extrapolating the last two minimizers.

4. Results

4.1. First Model

Now, we consider the set of parameters a=100, b=-90, c=-100. For better condition, we write the regression model as a + b exp(0.001ct).

For observations we choose the interval [030].

Here we assume that the deviation of the error is proportional to the measured value, the rate being constant 8%. So, at the left and, the mean error is 0.8, at the right end it is around ten times bigger, see Fig. 2.

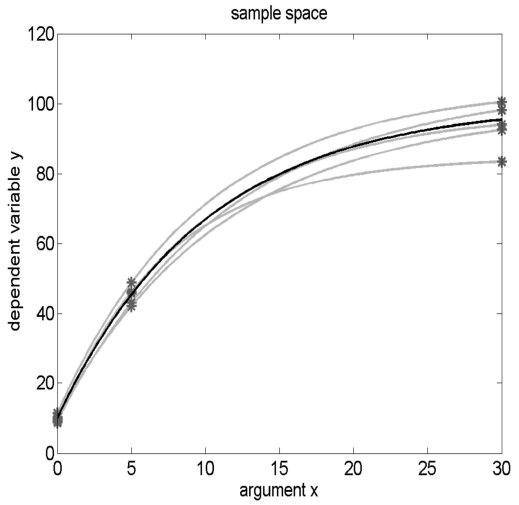

Fig. 2. Nonlinear regression model – other parameters

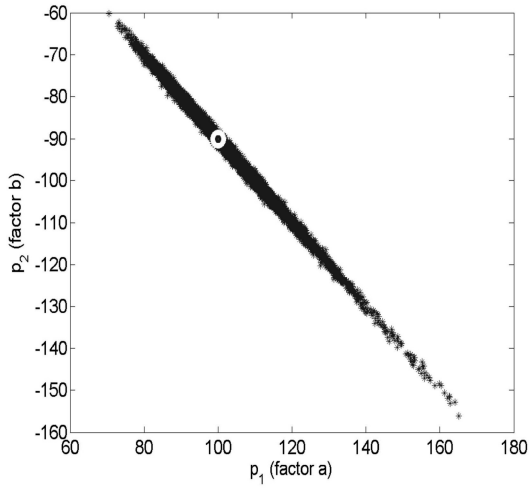

Fig. 3. Estimates for a and b

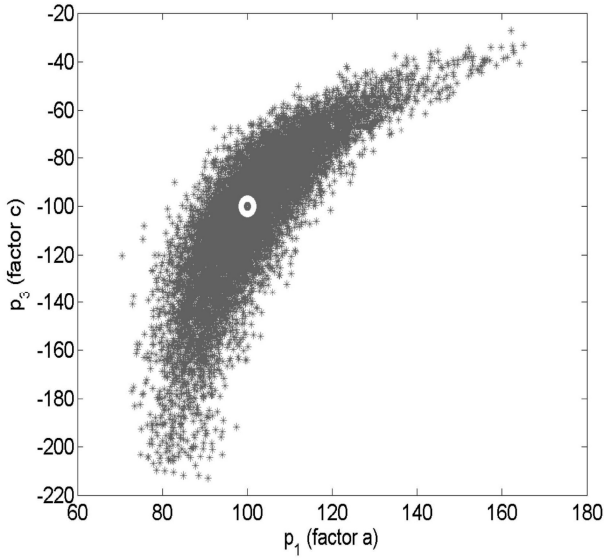

Fig. 4. Estimates for a and c

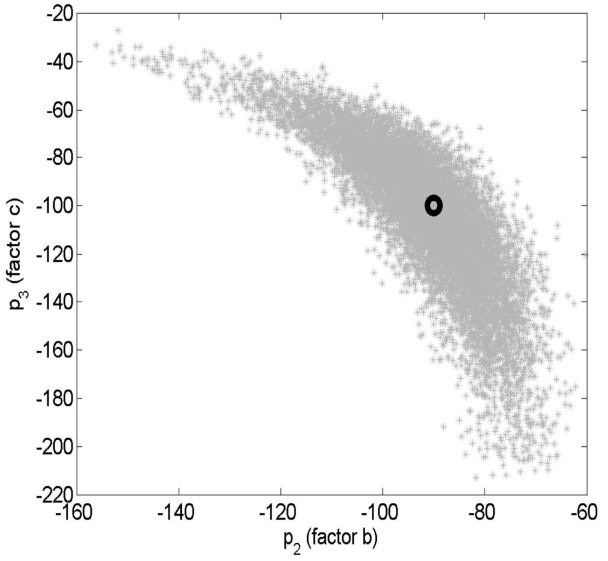

Fig. 5. Estimates for b and c

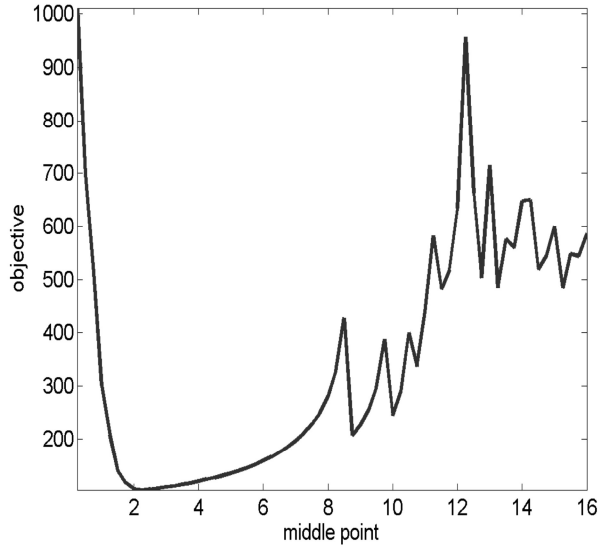

Fig. 6. Objective – diameter of parameter cloud

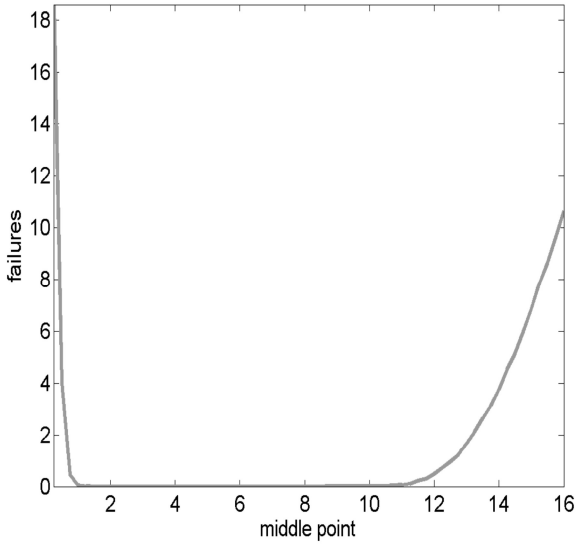

Fig. 7. Failure rate

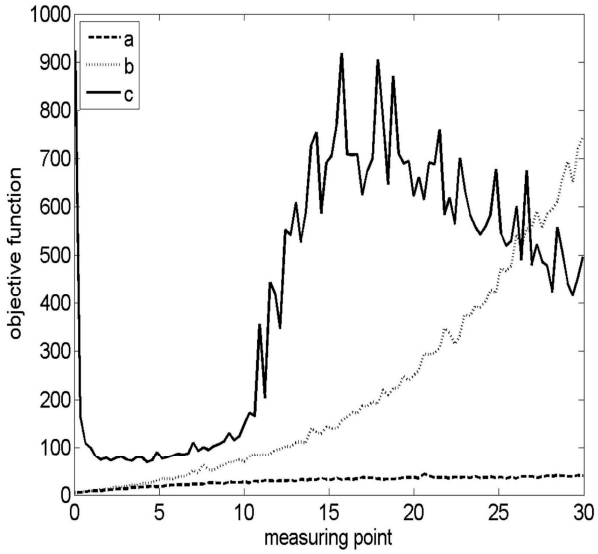

Fig. 8. Objective function for only measuring a, b or c

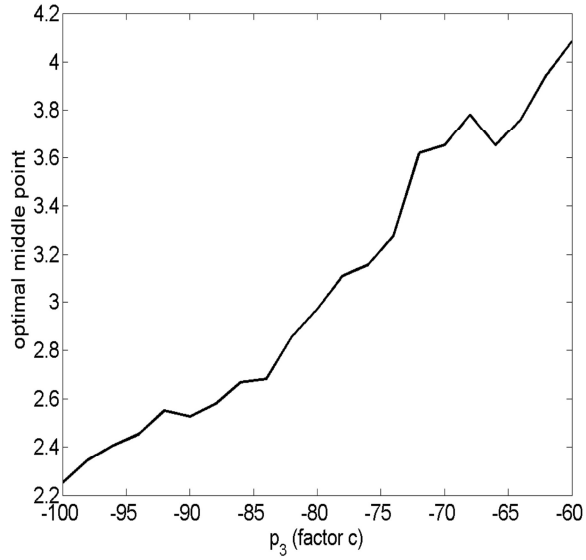

Fig. 9. Solution path of the homotopy optimization method and of the D-optimal design

The above results were obtained for 10000 trials, using a relative deviation of 0.08. Two measuring points were frozen at the ends of the observation interval, the middle point was varied.

It is obvious that for all parameters the nonlinear regression model is monotonous (in the time variable). For our case, the coefficient under the exponential function is negative, i.e., the curve tends to an asymptotic constant value.

For larger deviations (errors of the measurements) and measuring points near the right end of the observation interval, nonmonotonic measurements may occur. In such cases there is no (real) solution to the interpolation problem, i.e., the nonlinear system of 3 equations in 3 unknowns is a contradiction. We present the failure rate (percentage) in Fig. 7. Fortunately, in the region around the minimum of the objective, cf. Fig. 6, the failure rate is zero.

The oscillations of the objective for larger middle points are partially caused by our technique of cutting those failures. Rejecting a further quantile of outliers should leads to more regular results. We left the curve in its raw form since near the optimum it is smooth enough anyway.

The value of the objective function for only measuring one parameter is shown in Figure 8. When we only search for c, the minimizer is nearly the same as the optimal middle point for measuring all three parameters. In the other two cases, the diameter of the cloud gets minimal at the left end of the observation interval.

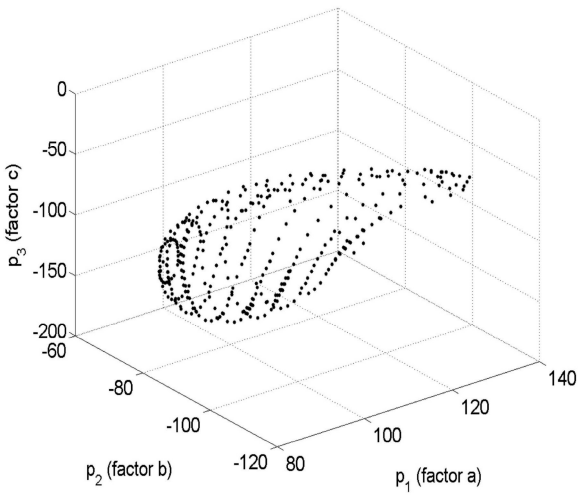

Fig. 10. Parameter cloud in 3D

The solution curve of the optimal middle point of our homotopy optimization method is presented in Fig. 9. We start with c = -100 and a known optimal point $x = (0, 2.25, 30)^T$. We chose a step size of 2 and the identity predictor for following

the path for c ∈ [-100, -60]. The obtained path shows an almost linear dependence. The optimal measuring point in the middle of the observation interval of the D-optimal design alsways lies right of our computed optimum, but the distance between both points is for all c ∈ [-100, -60] nearly the same.

4.2. Second Model

In our first example we set a = 1, b = 1, c = 3 and the observation interval [0,1]. We choose the same deviation of 0.5 for every point of the interval. So the relativ error at the left end is larger than at the right end.

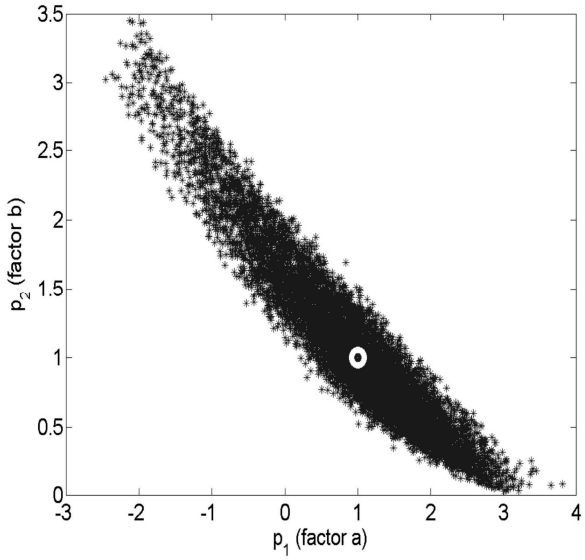

Fig. 11. Estimates for a and b

In Figure 14 the objective function has a local minimum at x ≈ 0.7 where x is the measuring point in the middle of the observation interval. The failure rate presented in Figure 15 is zero only between 0.4 and 0.9. so that we obtain a unique minimum in [0.4, 0.9]. If we only want to measure one parameter, the resulting objective functions are shown in Figure 16. When we are interested in b or c, we have to measure at the right end of the interval, but the objective for a has no minimum.

We perform our path following method for c ∈ [3, 7] with the known local minimizer $x^* = (0, 0.7, 1)^T$ and use the identity predictor. The resulting solution path for the optimal middle point as well as the D-optimal design are presented in Figure 17.

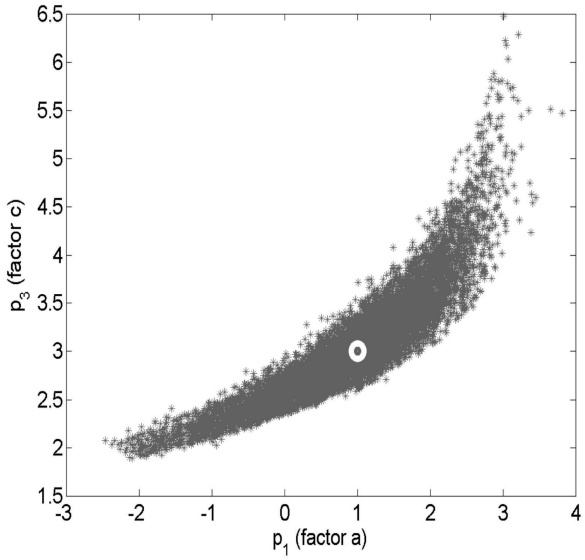

Fig. 12. Estimates for a and c

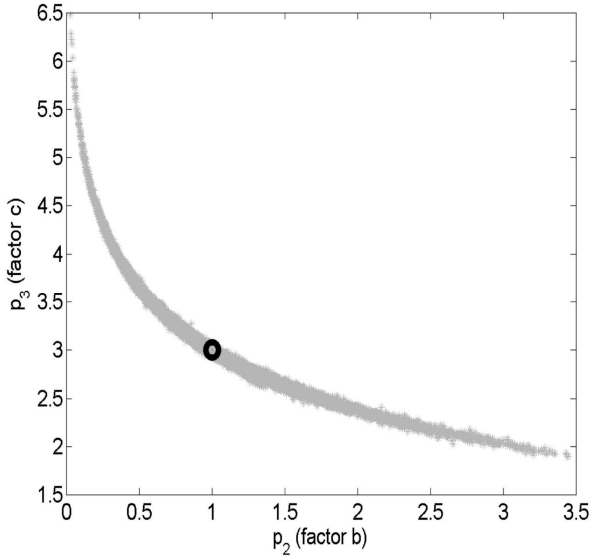

Fig. 13. Estimates for b and c

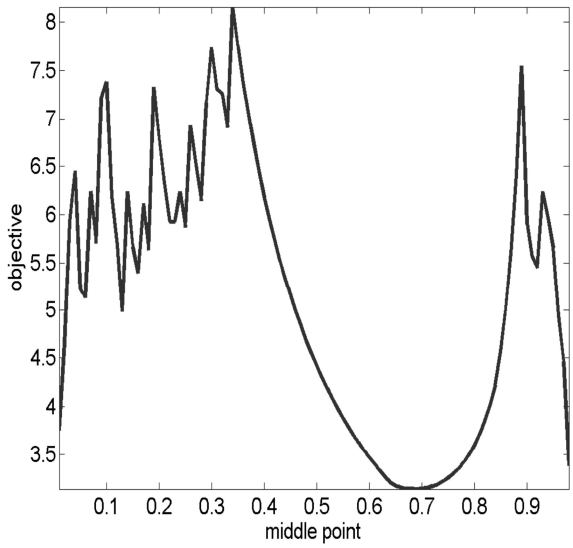

Fig. 14. Objective function

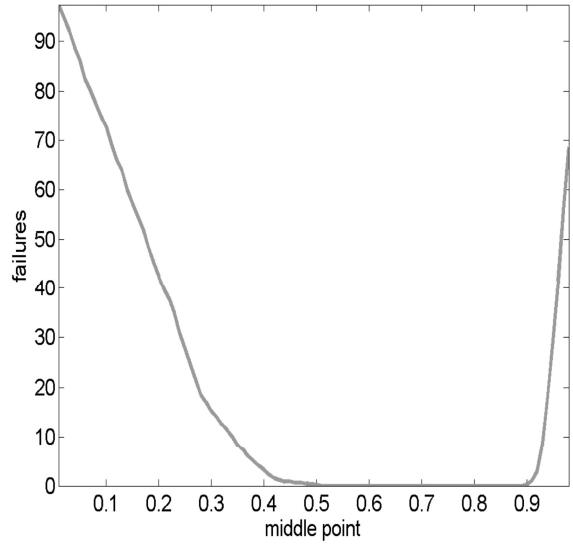

Fig. 15. Failure rate

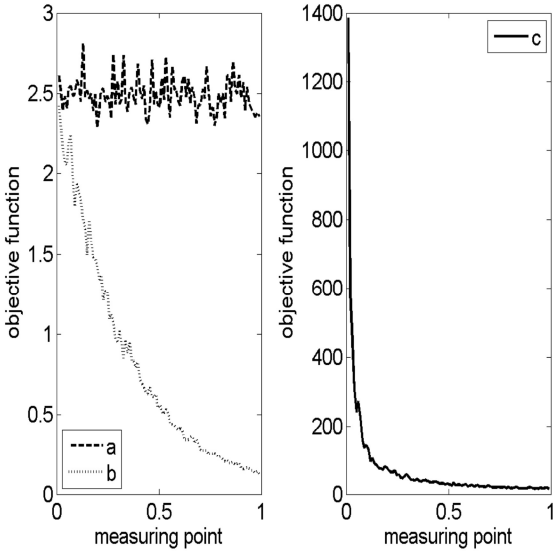

Fig. 16. Objective function for only measuring a, b or c

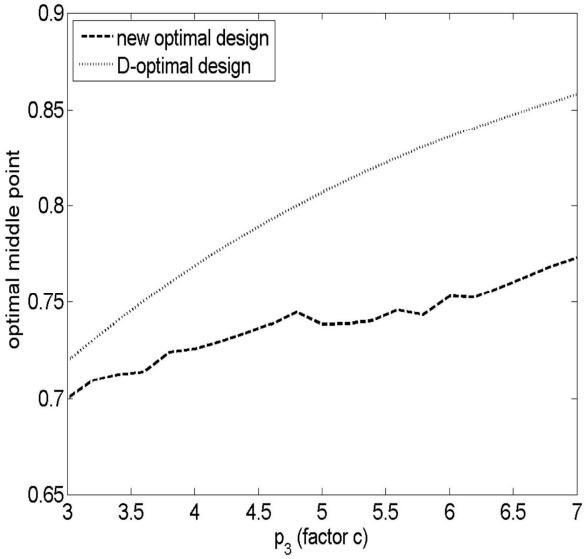

Fig. 17. Solution path of the homotopy optimization method and of the D-optimal design

4.3. Model with 5 Parameters

Now we choose a model $f(x, p) = a + b_1 e^{\frac{c_1}{1000}x} + b_2 e^{\frac{c_2}{1000}x}$ with the five parameters a = 100, $b_1 = -90$, $c_1 = -100$, $b_2 = -10$ and $c_2 = -350$. The deviation of 0.1% is proportional to the function values. The diameter of the parameter cloud is minimal if the measuring points are $x = (0, 0.1, 3.2, 16.2, 30)^T$.

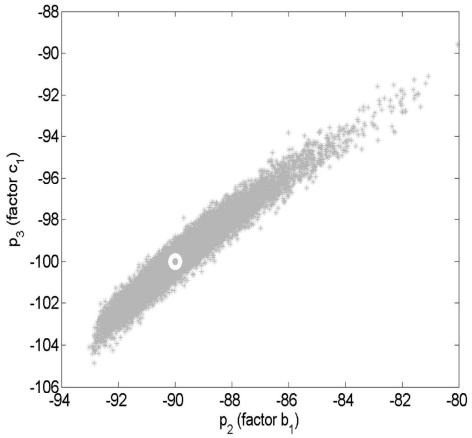

Fig. 18. Parameter cloud for b_1 and c_1

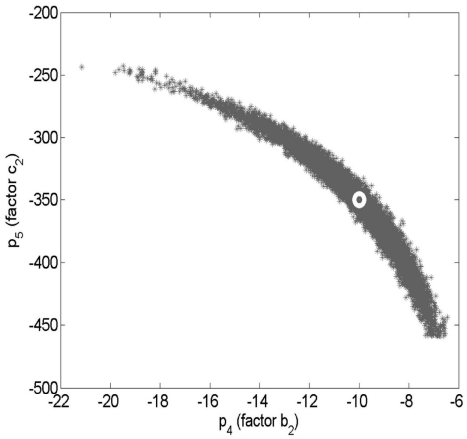

Fig. 19. Parameter cloud for b_2 and c_2

First, the parameter c_1 was varied and the solution paths for the three measuring points within the interval [0,30] are shown in Figure 20.

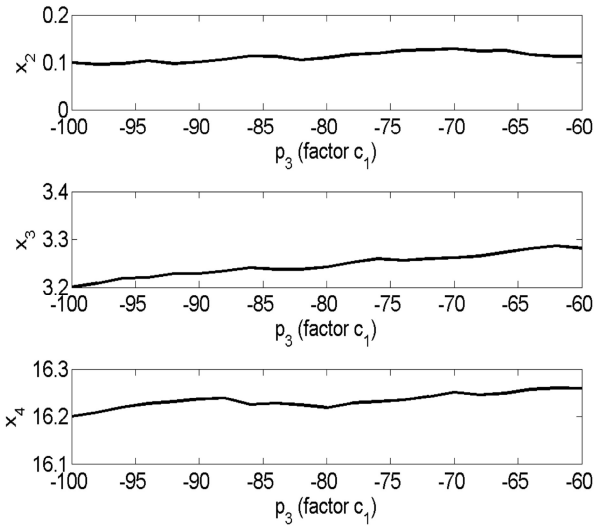

Fig. 20. Solution path of the homotopy optimization method for c_1

We then varied fifth parameter c_2 and the resulting solution curves are presented in Figure 21.

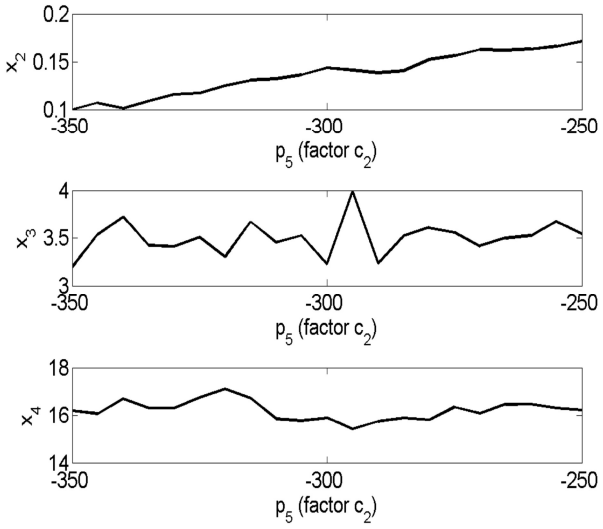

Fig. 21. Solution path of the homotopy optimization method for c_2

5. Conclusions

The presented approach to planning measurements is based on an assessment of stochastic errors in the parameters to be identified, which is obtained from computer simulations with randomly generated data. It is very flexible with respect to assumptions as to the error distribution in sample space, and much more realistic than classical approaches using the Fisher information matrix, however, at the price of solving a huge amount of least-squares problems. The random optimality functional obtained this way is not suitable for minimizing by gradient-type methods, thus a direct search is applied. To speed up calculations in the case of a systematic search of optimal designs over the whole parameter space an embedding method is used, by which we generate suitable initial guesses for each minimization process. A comparison with the classical approach is performed for a simple 3-parameter model. First results for a present application featuring a 5-dimensional independent variable round up the paper.

References

1. Box G.E.P., Lucas H.L.: Design of experiments in nonlinear estimations, Biometrica 49, 77-99, 1959.
2. Duchrau P., Frischmuth K., Rasch D., Schimke E.: Optimum experimental designs in growth curve analysis, Rostock. Math. Kolloq., 30, 93-104, 1986.
3. Frischmuth K.: An analytical result in experimental design, Rostock. Math. Kolloq., 31, 95-101, 1987.
4. Frischmuth K.: On locally D–optimal experimental designs, Rostock. Math. Kolloq., 36, 51-64, 1989 [5] P. Duchrau, K. Frischmuth, A numerical search algorithm for experimental design, Rostock. Math. Kolloq., 40, 83-94, 1990.
5. Frischmuth K., Hanler M.: On the inverse problem for the Ekman equation, ZAMM, 76, 9, 541-544, 1995.
6. Nelder J.A., Mead R.: A simplex method for function minimization, The Computer Journal 7, 308-313, 1965.
7. Langarias J.C., Reeds J.M., Whright M.H., Wright P.E.: Convergence properties of the Nelder-Mead simplex method in low dimensions, SIAM Journal on Optimization, 9, 1, 112-147, 1998.
8. Krantz S.G., Parks H.R.: The implicit function theorem: history, theory and applications, Birkhaeuser, Boston, 2003.
9. Uhl T.: Uncertainty and variability problems in rail system simulations, 10^{th} RWM Workshop, Rostock, March 30-31, 2008.