Model-based Design Analysis and Yield Optimization

Tobias Pfingsten, Daniel J.L. Herrmann, and Carl E. Rasmussen

Abstract— Fluctuations are inherent to any fabrication process. Integrated circuits and micro-electro-mechanical systems are particularly affected by these variations, and due to high quality requirements the effect on the devices' performance has to be understood quantitatively. In recent years it has become possible to model the performance of such complex systems on the basis of design specifications, and model-based Sensitivity Analysis has made its way into industrial engineering. We show how an efficient Bayesian approach, using a Gaussian process prior, can replace the commonly used brute-force Monte Carlo scheme, making it possible to apply the analysis to computationally costly models. We introduce a number of global, statistically justified sensitivity measures for design analysis and optimization. Two models of integrated systems serve us as case studies to introduce the analysis and to assess its convergence properties. We show that the Bayesian Monte Carlo scheme can save costly simulation runs and can ensure a reliable accuracy of the analysis.

I. Introduction

Before computational power was widely available, only relatively simple models were used in the design process of technical systems. Simple models require a deep understanding of the considered system as they need to be based on very specific assumptions, and one has to identify beforehand what question they are to answer. Today, however, simulation techniques exist which model all relevant features of a system, including geometrical properties, electrical and thermal aspects (finite element models) and circuit simulations. Such models are constructed as one-to-one emulations using powerful simulation tools, and do not necessarily lead themselves to a better understanding of the system—they are designed as virtual counterparts of the real system and mimic their behavior. The purpose of such models is to replace experimental specimens which can be extremely expensive and time-consuming in their fabrication.

Variations during production are inevitable in any process. Especially in the fabrication of integrated systems, these tolerances may not be small in comparison to the dimensions of the devices and may thus have a significant effect on the functionality of the product. Therefore the performance of the product will vary from device to device. High quality standards require the characteristics of a product to lie within a small tolerance window. Therefore the main task of simulation in this field is to assess and optimize the robustness of the design against processes inherent variations.

Computer models which simulate the behavior of a device can be seen as a deterministic mapping from the specifications of the system (input parameters) to its properties (output parameters). The techniques to estimate the variation in the output are studied in the so-called *Uncertainty* or *Sensitivity Analysis*, by combining the deterministic computer model and the distributions of the input parameters.

The most direct way to do Sensitivity Analysis is the Monte Carlo (MC) method. MC is very simple to implement but may be computationally costly: MC draws a number of random input parameters from their distribution and runs the simulation code on each setting. The outputs resemble the attributes of a family of devices which leave the production line. If the computational cost of the computer model is small the model can be evaluated at a great number of parameter settings and MC is the method of choice. However, if one run of a complex model takes several minutes or even hours, we need to avoid an excessive number of runs. Furthermore, an MC analysis has to be repeated a number of times in order to calculate sensible measures. The gathered data can hardly be re-used if the settings change.

In this work we present a new technique which uses the available simulation runs more efficiently than MC approaches. The method lets the designer re-use the data in a consequent analysis. This is achieved by applying a state-of-the-art algorithm from Machine Learning to learn an emulation of the computer model. While the methods for function estimation are well-known, our contribution is their adaption to design analysis and optimization. We define statistically justified measures for the sensitivity of designs, which ease the assessment of high-dimensional, nonlinear models. The sensitivity measures can be computed in closed form for most relevant input distributions—including derivatives with respect to specifications. This makes it possible to use them in an automatic design optimization.

In the spirit of a method first introduced by Haylock and O'Hagan [7], we describe a Bayesian Sensitivity Analysis which uses a Gaussian process (GP) prior. GPs are long known as flexible nonparametric models for high-dimensional regression [26]. They have been used as fast surrogates (response surfaces) of expensive computer code in a number of previous works [2], [7], [16], [17], [25], [32], [22]. By using GP models in our evaluation, we assume that the simulation output is a smooth function of the input parameters.

Tobias Pfingsten and Daniel J.L. Herrmann are with Robert Bosch GmbH, Stuttgart. Carl E. Rasmussen is with the Max Planck Institute for Biological Cybernetics, Tübingen.

The Bayesian scheme is separated into two independent steps. Using previous simulation runs we build an a-posteriori Gaussian process model of the computer code in a first step. The analysis, as a following second step, is independent of the computer model itself and uses only the fast GP surrogate for the model. Therefore the analysis can be done in little time and independently of expensive simulation software, thus saving resources and labor time. The GP meta-model can be tested using e.g. a cross validation scheme, facilitating an improved reliability of the analysis.

Antreich et al. [1] and Sapatnekar et al. [29] propose geometric optimization methods for design-centering, which approximate the feasibility region and the input distributions. In contrast, our approach treats the fluctuations of the input parameters exactly to define appropriate sensitivity measures. It uses the approximative GP model to compute them efficiently. Response surface methods (RSM) [13] are strongly related to our approach. Previous works reporting on design optimization with RSM [8], [5] use linear or quadratic fits. Dealing with complex models, these parametric fits can only be used as local approximations. Our approach generalizes the idea by using nonparametric GP regression, which can be used to estimate the function globally.

To evaluate the method we use two fully featured models of electro-mechanical systems. Our high-dimensional examples represent a pressure sensor and an accelerometer, which are defined by 28 and 29 uncertain input parameters. We show that the proposed scheme can lead to a significant saving of simulation runs, and that it can help to re-use valuable data in several design studies.

In Section II we address the definition of sensitivity measures for a model-based assessment of designs. We use a real-world design study to exemplify the proposed design analysis. The Monte Carlo approach is the standard method to compute these quantities. We propose to use instead a Bayesian quadrature[1] using Gaussian process regression, and describe both in Section III. We compare the efficiency of the Bayesian approach to the classical MC method with Latin-Hypercube sampling. In Section IV we compare the convergence rates in an empirical analysis, using the mentioned sensor models and a synthetic benchmark problem. Section V closes with a synopsis of the results.

II. SENSITIVITY ANALYSIS FOR DESIGN ASSESSMENT

Sensitivity Analysis addresses the qualitative (screening) and quantitative study of how influential input parameters are in nonlinear mathematical models. A Sensitivity Analysis is often performed to assess the stability of a system with respect to small disturbances. In design analysis, however, the fluctuations in the parameters are not necessarily small.

One distinguishes *local* and *global* measures for sensitivity [27], [28]. Local measures are mostly based on partial derivatives or use some kind of parametric response surface (linear or quadratic approximation). We can certainly use local measures if we deal with small disturbances. However, we are forced to

---

[1]Quadrature is the classical term for approximate integration.

use a global criterion if the support of the input distribution $p(\mathbf{x})$ is not small enough to neglect high order terms.

Accordingly Saltelli [27] defines a global method to have two basic properties: The inclusion of the influence of the scale and shape of $p(\mathbf{x})$ and a multidimensional averaging by a joint variation of all parameters.

In II-A we propose a number of global measures which are useful in design analysis. To motivate these, we start by introducing the *standardized regression coefficients* as a prominent local measure. In II-B we show how the Sensitivity Analysis can be used in practice, reproducing the analysis for the electro-mechanical pressure sensor.

A. Measures for the robustness of a design

a) Local measures: For underlying models which are approximately linear over the support of $p(\mathbf{x})$,

$$f(\mathbf{x}) \approx f_{\text{lin}}(\mathbf{x}) = a_o + \sum_\ell a_\ell x_\ell , \qquad (1)$$

the standardized regression coefficients (SRCs) [27] are a common measure for sensitivity. Under the assumption that the inputs are normally distributed, $p(\mathbf{x}) = \prod_\ell p_\ell(x_\ell) = \prod_\ell \mathcal{N}(x_\ell | \hat{x}_\ell, \sigma_\ell^2)$, the output distribution $p_{\mathbf{x}}(f_{\text{lin}})$ is also normal. The variance can be decomposed into independent contributions from each input parameter: $\text{var}_{\mathbf{x}}[f_{\text{lin}}] = \sum_\ell a_\ell^2 \sigma_\ell^2$. The SRCs are defined as the relative shares of the variance due to fluctuations in single parameters:

$$\text{SRC}_\ell = \frac{\text{var}_{\mathbf{x}}[f_{\text{lin}} | \text{fix all inputs but } x_\ell]}{\text{var}_{\mathbf{x}}[f_{\text{lin}}]} = \frac{a_\ell^2 \sigma_\ell^2}{\text{var}_{\mathbf{x}}[f_{\text{lin}}]} . \quad (2)$$

A linear fit of f can be computed using few function evaluations, which makes the SRCs a popular and robust first estimate in model based sensitivity estimation. For the evaluation of MEMS it has been proposed by [31] and [4].

b) Global measures: According to [27] global measures are necessarily some kind of average over the joint distribution of input parameters $p(\mathbf{x})$:

$$I[f] = \int \mathrm{d}\mathbf{x}\, p(\mathbf{x})\, F[f(\mathbf{x})] , \qquad (3)$$

where F is some functional of the output f.

The parametric yield is certainly the most important measure for the robustness of a design with respect to fluctuations in the manufacturing process. It is by definition the fraction of outputs $f(\mathbf{x})$ which, according to $p(\mathbf{x})$, lies within a given tolerance interval $I_{\text{tol}} = [f_{\text{min}}, f_{\text{max}}]$:

$$\text{Yield}[f] = \int \mathrm{d}\mathbf{x}\, p(\mathbf{x}) \begin{cases} 1 & \text{for } f(\mathbf{x}) \in I_{\text{tol}} \\ 0 & f(\mathbf{x}) \notin I_{\text{tol}} \end{cases} \qquad (4)$$

Using the parametric yield we can immediately define a measure of impact for single input parameters. We define the yield gain as the increase in the parametric yield, obtained by a perfect control of a parameter to its nominal value $\hat{x}_\ell$:

$$\Delta \text{Y}_\ell = \text{Yield}[f | \text{where } x_\ell = \hat{x}_\ell] - \text{Yield}[f] . \qquad (5)$$

If the output distribution $p_{\mathbf{x}}(f)$ is approximately normal, it can be summarized by its mean and variance which are given by the integrals

(a) Linear and nonlinear impact
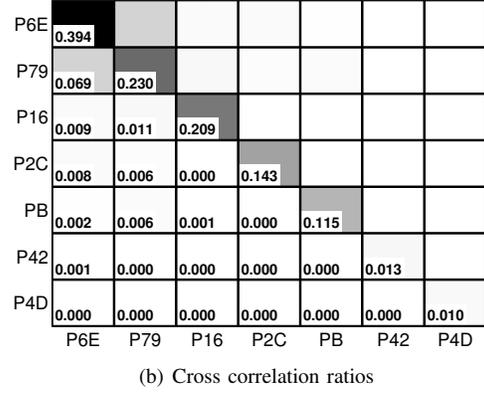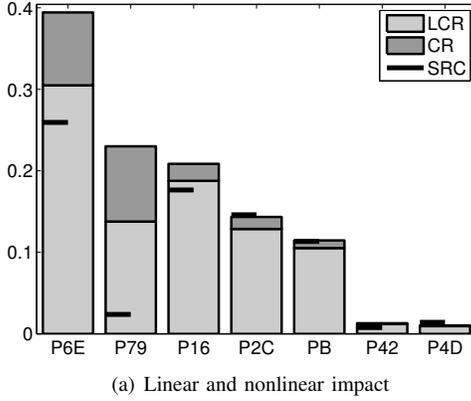

(b) Cross correlation ratios

Fig. 1. Design analysis (PS model) for the most influential design parameters. Panel (a): The correlation ratios CR include all effects, their local counterparts LCR are computed only along the axis. The SRC are based on a linear model. Panel (b): The matrix elements (numbers and gray shades) are the cross correlation ratios CCR. They measure the importance of mixed terms.

$$\text{mean}_{\mathbf{x}}[f] \quad = \quad \int \mathbf{dx}\, p(\mathbf{x})\, f(\mathbf{x}) \tag{6a}$$

$$\text{var}_{\mathbf{x}}[f] \quad = \quad \int \mathbf{dx}\, p(\mathbf{x})\, f^2(\mathbf{x}) \; - \text{mean}_{\mathbf{x}}[f]^2 \; . \tag{6b}$$

Variance based measures for sensitivity are widely used to measure the impact of input parameters.

As a generalization of the SRCs (2) to nonlinear mappings we define the coefficients

$$\text{CR}_\ell = \frac{\text{var}_{\mathbf{x}}\,[f] - \text{var}_{\mathbf{x}}\,[f|x_\ell = \hat{x}_\ell]}{\text{var}_{\mathbf{x}}\,[f]} \; , \tag{7}$$

where $\hat{\mathbf{x}}$ is the nominal parameter setting. We will refer to these parameters as the *correlation ratios* (CR), in allusion to a similar definition in [12]. Just as the SRCs, the CR_ℓ measure by which fraction the variance of f can be reduced by a perfect control of x_ℓ. Oakley and O'Hagan [17] use slightly different measures, as they consider cases in which no sensible nominal value can be defined.

The CRs measure the average influence of a single parameter over the complete distribution $p(\mathbf{x})$. When the inputs are not correlated, i.e. $p(\mathbf{x}) = \prod_\ell p(x_\ell)$, the CRs are independent of the fluctuations in other parameters if f is additive, $f(\mathbf{x}) = \sum_\ell f_\ell(x_\ell)$, or even linear. However, mixed terms can lead to a mutual interaction of the parameters. Consider, for example, the term $\mathbf{x}_i\mathbf{x}_\ell$. For $\mathbf{x}_i = \hat{\mathbf{x}}_i = 0$ the impact of $\mathbf{x}_\ell$ is zero, for fluctuating $\mathbf{x}_i$, however, not necessarily.

To measure the importance of these *cross terms* in the CRs, we define the *local correlation ratios*

$$\text{LCR}_\ell = \frac{\text{var}_{\mathbf{x}}\,[f|x_i = \hat{x}_i \quad \forall\, i \neq \ell]}{\text{var}_{\mathbf{x}}\,[f]} \; , \tag{8}$$

which do not include such terms: The average is taken only along the axis of the considered parameter. To be able to assign the cross terms to pairs of parameters, we define *cross correlation ratios* (CCRs)

$$\text{CCR}_{i\ell} \quad = \quad \text{CCR}_{\ell i} = \text{CR}_i - \text{CR}_i\,(x_\ell = \hat{x}_\ell) \; , \tag{9}$$

which reveal how strongly two parameters are linked. The definition (9) is intuitively clear: It measures the change of CR_i as we fix parameter x_ℓ to its nominal value. Where mixed terms can be neglected, we have $\text{CR}_i = \text{CR}_i\,(x_\ell = \hat{x}_\ell)$ and

the cross terms are zero. If two parameters x_i and x_ℓ are maximally correlated through f, we have $\text{CCR}_{i\ell} = \text{CCR}_{\ell\ell} = \text{CR}_\ell = \text{CCR}_{ii} = \text{CR}_i$.

In the analysis of our designs we use plots which combine all four measures, SRC (2), CR (7) , LCR (8) and CCR (9) to indicate the effects which dominate the model.

B. Case study: Use in practice

To motivate the use of the above sensitivity measures we exemplify the Sensitivity Analysis in the following paragraph. We have performed the analysis using the Bayesian Monte Carlo Method, which we explain in detail in Section III. The case study reproduces the model-based design analysis of an electro-mechanical pressure sensor, in development at Robert Bosch GmbH. Note that we have anonymized the model by normalizing outputs and renaming all input parameters.

1) Model 1, Pressure Sensor: The model stems from the design analysis of a pressure sensor (PS) and covers all relevant mechanical and electrical properties of the system. A finite element model of the mechanical configuration reproduces the deformation of the device due to the applied pressure. The mechanical module has a number of parameters which represent the geometrical dimensions. The deformations are in turn converted into electrical signals. The output of the model is a temperature and pressure dependent electrical signal, for which a last module calculates significant characteristics such as the accuracy of the device. The model has 28 parameters in total, for which typical fluctuations in the production process are known.

One model run requires approximately 2min on a 2GHz CPU. An exhaustive MC analysis requires thousands of function evaluations and can therefore not be under consideration for a variety of design alternatives. According to the probabilistic error bound of the MC method we would need 5000 samples for an accuracy of 1.5% in the mean estimate.

Instead, the Bayesian Monte Carlo approach uses a Gaussian process meta-model which is trained and tested on comparably few simulation runs. We have used a 500 points-Latin Hypercube design from $p(\mathbf{x})$ for training. To calculate an estimate of the model accuracy we have used a separate test set of

(a) Mean output and variation vs. nominal P6E   (b) Mean output and variation vs. nominal P79   (c) Output variation against P6E and P79
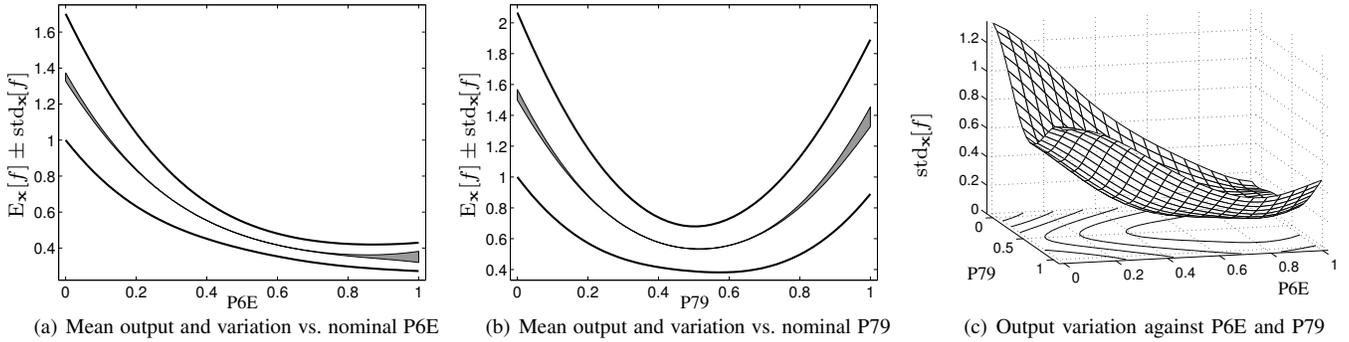
Fig. 2. Design study (PS model): The plots (a) and (b) show the robustness of the design for varying nominal parameters P6E and P79. Shown are the mean output with $\pm\sigma$ confidence interval (shaded area) and the output variation (thick lines at $\pm\mathrm{std}_\mathbf{x}[f]$). Plot (c) shows the output variation $\mathrm{std}_\mathbf{x}[f]$ against nominal values for P79 and P6E. Note that the settings can be adjusted as to minimize the global uncertainty.

1000 samples. Note that a test set is not necessary if cross validation is used. The square root of the mean square error on the test set was 1.3% of $\mathrm{std}[f(\mathbf{X}_{\mathrm{test}})]$, thus ensuring a good accuracy of the GP-meta model. One can easily verify that the accuracy of the estimated CRs is of the same order as the maximal squared error of the regression model.

2) Verification of a fixed design: The plots in Figure 1 combine several sensitivity measures to provide a condensed and comprehensive picture of the parameters' role. Since the Bayesian Sensitivity Analysis uses a fast GP emulator of the computer code, we can calculate these characteristics for varying designs within seconds.

The difference between the CRs and SRCs indicates the importance of nonlinear effects to the design engineer. Nonlinear effects are apparently responsible for most variation caused by parameter P79 in the PS model. The lower part of the bars, shaded in a lighter gray, shows the local correlation ratios LCR. The difference to the global CRs indicates what fraction of the impact is induced by cross effects or dependencies to other parameters.

These cross effects are broken down by the CCR matrix, which is shown in plot 1(b). The CRs can be found on the diagonal and the CCRs are placed off the diagonal, indicating the interdependencies between pairs of parameters. Note, for example, that the first two parameters, P6E and P79, affect the output jointly.

The above variance measures are based on the assumption that the output distribution is approximately normal. In general these coefficients only estimate the width of the distribution. The parametric yield (4) or the potential yield improvements ΔY (5) can be computed to assess the absolute sensitivity of the design.

3) Parameter studies and design optimization: The mean output and the output variation depend on the tolerances of the input parameters as well as on their nominal values. The dependence of the output variation on the design parameters is frequently studied in the designing process. The design parameters can often be chosen freely within some interval and should be set to values where the output fluctuations are minimal. Tolerances are an important expense factor in the manufacturing process, and therefore it is also worthwhile to study their effect on the output fluctuations quantitatively.

Using the MC method, one complete analysis has to be performed for each setting of design parameters, nominal values, and tolerances. The GP regression, which we propose to use as an intermediate step, lets the developer investigate such dependencies independently of the computer model without the need to wait for simulation runs.

Figure 2 shows exemplary parameter studies for the PS model. In 2(a) and 2(b) we have plotted the mean output, the predictive uncertainty, and the output fluctuations against the nominal parameter. In agreement to the large difference between the SRC and CR given in Figure 1, we find the dependencies to be highly nonlinear. The output variation is not constant over the parameter settings, and we find in plot 2(c) that it can be minimized by adjusting the nominal values.

As all parameters of the BMC analysis are given as analytically manageable functions, we can derive all necessary derivatives to use a gradient based optimization scheme.

III. BAYESIAN MONTE CARLO USING GAUSSIAN PROCESS PRIORS

In the above section we have derived and motivated a number of global sensitivity measures. In the following we show how these can be computed accurately from few simulation runs. We briefly introduce the traditional MC method in III-A, which uses simple estimates. The Bayesian Monte Carlo method, which makes more efficient use of the data by using a Gaussian process prior, is presented in III-B.

A. Monte Carlo Methods

In general—especially if f is represented by a complex computer code—integrals of type (3) cannot be evaluated analytically and we have to resort to numerical approximations. Classical methods like the trapezoidal rule are not applicable for high dimensional input spaces $\mathbb{R}^D$: The quadrature error scales as $\mathcal{O}(N^{-2/D})$—for $F \in \mathcal{C}^2$ and N being the number of nodes. Monte Carlo (MC) methods, in contrast, lead to a probabilistic error bound of $\mathcal{O}(N^{-1/2})$ which is independent of the input dimension [15].

The basic idea of Monte Carlo methods is to draw a finite number of N samples $\mathbf{x}_1 \ldots \mathbf{x}_N$ from $p(\mathbf{x})$ and to use the
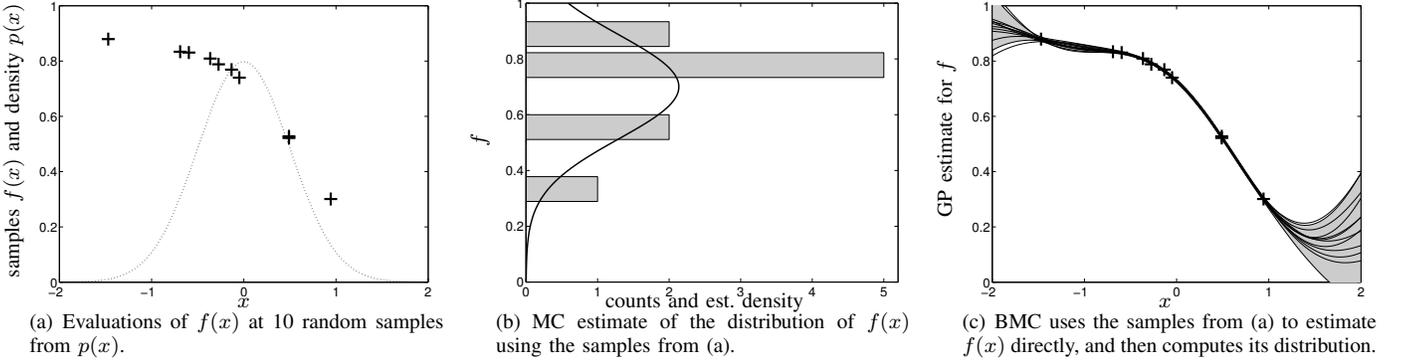
(a) Evaluations of $f(x)$ at 10 random samples from $p(x)$.

(b) MC estimate of the distribution of $f(x)$ using the samples from (a).

(c) BMC uses the samples from (a) to estimate $f(x)$ directly, and then computes its distribution.

Fig. 3. Univariate illustration. Both, MC and BMC estimate $E_{\mathbf{x}}[f]$ and $\mathrm{var}_{\mathbf{x}}[f]$ from evaluations of f at single points x. Panel (a) shows $f(x)$ at 10 random samples '+' from a Gaussian input distribution $p(x)$ '$\cdots$'. The MC method uses the samples $f(x)$ via empirical estimates (10). See the corresponding histogram and the empirical Gaussian approximation to $p_{\mathbf{x}}(f)$ '—' in (b). In contrast, BMC uses the samples to directly estimate f and computes $\mathrm{mean}_{\mathbf{x}}[f]$ and $\mathrm{var}_{\mathbf{x}}[f]$ on the basis of this approximation. Panel (c) shows the samples from (a) '+', with the 2σ confidence interval for f (gray) and function samples from the posterior '—'.

empirical mean

$$ I[f] \quad \approx \quad \frac{1}{N} \sum_{\ell} F[f(\mathbf{x}_\ell)] \qquad (10) $$

as an unbiased estimator of the expectation in (3). The average error and the probabilistic bound are guaranteed by the strong law of large numbers and the central limit theorem, independently of $F[f]$.

In the simplest case independent random samples from $p(\mathbf{x})$ are used, more sophisticated *Quasi Monte Carlo* methods use e.g. Latin Hypercube designs [11] or Sobol lattices [30]. Quasi Monte Carlo methods basically lead to improved space filling. In this paper we use the Latin Hypercube technique.

The error bound $\mathcal{O}(N^{-1/2})$ for MC methods holds for a very broad class of functions, requiring only square-integrability. While this can be seen as an advantage, it is clear that for highly regular functions relatively few nodes should be necessary to approximate the integral. Therefore it can be worthwhile to reflect this regularity in a quadrature rule.

B. Bayesian Monte Carlo

Monte Carlo methods—including improved Quasi MC methods like Latin Hypercube—directly estimate the averages by empirical sums (10). The Bayesian MC method uses an indirect estimate, where the underlying function f is modeled using a GP.

GPs have repeatedly been proposed as a replacement for parametric response surfaces: As GPs are nonparametric models, they can adapt to the output most of computer models over a large range of input parameters. Using the GP model, the available simulation data can be used efficiently to approximate the function. All measures can subsequently be computed using this approximation:

Bayesian Monte Carlo

A   *Generate Data.* Choose a design $\mathbf{X} = (\mathbf{x}_1 \ldots \mathbf{x}_N)^T$ in the region of interest and evaluate the function f at these inputs: $\mathbf{y} = (f(\mathbf{x}_1) \ldots f(\mathbf{x}_N))^T$.

B   *GP regression.* Use the data $\mathcal{D} = \{\mathbf{X}, \mathbf{y}\}$ to calculate a GP fit $p(f|\mathcal{D})$. The model can be verified using a separate test set or cross validation.

C   *Sensitivity Analysis.* Any analysis can be performed using the GP emulation of f. The integrals $I[f]$ (3) are estimated using the GP fit.

The underlying idea is relatively general and applies also to other frameworks: The computer code is emulated efficiently using the GP model, which can be used for all further analysis.

Find a simple univariate example in Figure 3. Panel (a) displays 10 evaluations of a function $f(x_\ell)$ at inputs $x_1 \ldots x_{10}$, which are randomly drawn from $p(x)$. The MC estimate (panel b) uses the samples $f(x_\ell)$ to compute $\mathrm{mean}_{\mathbf{x}}[f]$ and $\mathrm{var}_{\mathbf{x}}[f]$ directly. The BMC method uses the given data to estimate the function f. In panel (c) we have displayed this estimate by showing the 2σ confidence interval for f together with samples from the posterior distribution. Using this approximation, $\mathrm{mean}_{\mathbf{x}}[f]$ and $\mathrm{var}_{\mathbf{x}}[f]$ can be computed analytically.

The following overview over GP models gives some insight into the basic assumptions and the mathematical framework. As GPs are now a well-known tool in Machine Learning, implementations of the described fitting procedure are available [21]. The complexity of the GP fit is $\mathcal{O}(N^3)$, and does not directly depend on the dimensionality of the input space D. Applications with hundreds of input parameters have been considered in the Machine Learning literature.

We explain BMC by going through each step of the method: In III-B.1 we introduce the GP model, and in III-B.2 we describe how the model is used to infer the underlying function from single measurements. III-B.3 explains how remaining parameters are fitted using the maximum likelihood method of type II.

Once the GP estimate to f is available, the sensitivity measures can be computed analytically (III-B.4). Note, that using the available software, an implementation of the BMC method requires only the coding of the equations (22) given in the appendix.

1) The Gaussian process model: GPs have been introduced by Sacks and Ylvisaker [26] to model systematic deviations from parametric models by considering correlated errors. O'Hagan [18] derived them as localized linear models. Both interpretations emphasize the nonparametric nature of GPs, which can overcome intrinsic problems of parametric
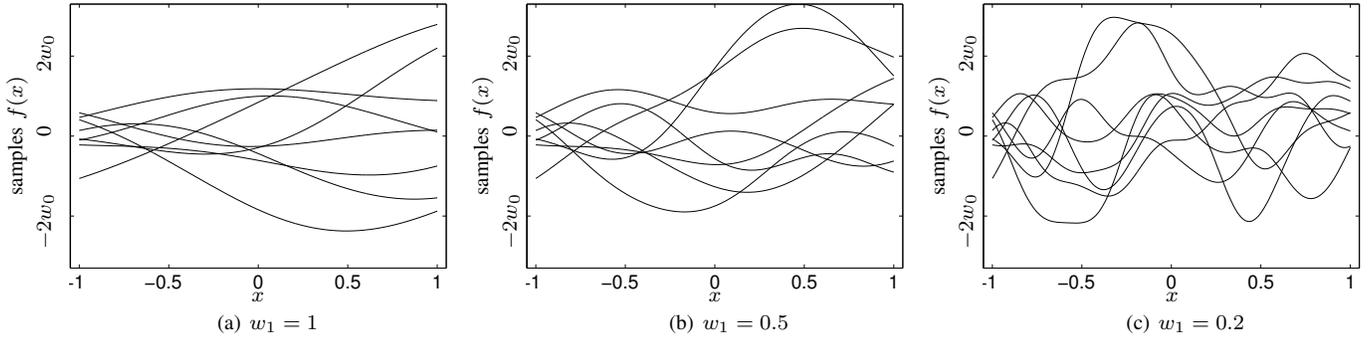
Fig. 4.   Random functions drawn from a one dimensional Gaussian process. We have used the covariance function (13) with different length scale parameters w_1. The plots illustrate the influence of the parameters w_o and w_1. Parameter w_o indicates the amplitude of the variations of $f(x)$. The length scale w_1 measures the smoothness of the function via the decay of correlations between the function values. The univariate example is equivalent to a multivariate model projected onto one axis. The GP model is not restricted to a finite number of basis functions, it models the function's smoothness via (13).

models. Parametric models are of limited use especially in high dimensional spaces—e.g. when complex computer code is modeled [25].

A comprehensive treatment of the GP model is out of the scope of this work, and we refer to [24] for more details. However, a Matlab implementation for GP regression is available electronically at [21], and the BMC method can therefore be implemented easily.

Assume we model a mapping f from the input parameters $\mathbf{x} \in \mathbb{R}^D$ to an output $f(\mathbf{x}) \in \mathbb{R}$. Gaussian processes extend parametric models by allowing for systematic deviations from a (parameterized, e.g. linear) mean function

$$\mathrm{E}[f(\mathbf{x})] = \mu(\mathbf{x}) . \tag{11a}$$

When the deviations from the mean function are systematic, they show correlations which are described by a fixed covariance structure

$$\mathrm{cov}\left[f(\mathbf{x}), f(\bar{\mathbf{x}})\right] = k(\mathbf{x}, \bar{\mathbf{x}}) . \tag{11b}$$

For notational simplicity we set the mean function to zero in the following, and concentrate on the structure of the deviations. Subsequently we discuss how to re-introduce the mean function to the Sensitivity Analysis.

GPs can be seen as a generalization of the well-known linear model [3]. In the following lines we briefly illustrate this connection to motivate the assumptions, which are introduced by the choice of the covariance function. In the linear model we assume

$$f(\mathbf{x}) = \sum_{\ell=1}^{k} \alpha_\ell \phi_\ell(\mathbf{x}) = \boldsymbol{\alpha}^T \boldsymbol{\phi}(\mathbf{x}) , \tag{12a}$$

where the $\boldsymbol{\phi}(\mathbf{x}) = (\phi_1(\mathbf{x}), \dots \phi_k(\mathbf{x}))^T$ are k basis functions, and the $\boldsymbol{\alpha} = (\alpha_1, \dots \alpha_k)^T$ are the corresponding coefficients.

The $\boldsymbol{\alpha}$s are usually assigned a Gaussian prior, $\alpha_\ell \sim \mathcal{N}(0, w_o^2)$, which leads to

$$\mathrm{E}_{\boldsymbol{\alpha}}[f(\mathbf{x})] = \mathrm{E}_{\boldsymbol{\alpha}}\left[\boldsymbol{\alpha}^T \boldsymbol{\phi}(\mathbf{x})\right] = 0 \tag{12b}$$

$$\mathrm{E}_{\boldsymbol{\alpha}}[f(\mathbf{x}), f(\bar{\mathbf{x}})] = \boldsymbol{\phi}(\mathbf{x})^T \mathrm{E}_{\boldsymbol{\alpha}}\left[\boldsymbol{\alpha}\boldsymbol{\alpha}^T\right] \boldsymbol{\phi}(\bar{\mathbf{x}}) \tag{12c}$$

$$= w_o^2 \boldsymbol{\phi}(\mathbf{x})^T \boldsymbol{\phi}(\bar{\mathbf{x}}) = \mathrm{cov}[f(\mathbf{x}), f(\bar{\mathbf{x}})] .$$

Identifying the covariance function with (12c), we see that the choice of the basis functions is reflected by the covariance function. For the special case $\boldsymbol{\phi}(\mathbf{x}) = \mathbf{x}$ we obtain $\mathrm{cov}[f(\mathbf{x}), f(\bar{\mathbf{x}})] = w_o^2 \mathbf{x}^T \mathbf{x}$.

The linear model is little flexible and it is hard to argue why the correlation between function values should increase with their distance. A common choice is to assume that correlations between the function values decay exponentially, i.e.

$$k(\mathbf{x}, \bar{\mathbf{x}}) = w_o^2 \exp\left\{-\tfrac{1}{2} \sum_{d=1}^{D} \left(\frac{x^{(d)} - \bar{x}^{(d)}}{w_d}\right)^2\right\} . \tag{13}$$

Here, it is assumed that function values at close inputs $\mathbf{x} \approx \bar{\mathbf{x}}$ are perfectly correlated, which corresponds to the assumption that the function is smooth. For large $|\mathbf{x} - \bar{\mathbf{x}}|$ we obtain $k(\mathbf{x}, \bar{\mathbf{x}}) \approx 0$, and the function values $f(\mathbf{x})$ and $f(\bar{\mathbf{x}})$ can be considered independent. It can be shown [24], that the covariance function in (13) corresponds to an infinite number of basis functions, which is why we call the model "non-parametric".

The parameter w_o controls the strength of the correlations, and the $w_1 \dots w_D$ are the typical length scales of the individual input dimensions. We collect the parameters in a vector $\boldsymbol{\theta} = (w_o \dots w_D)$. In Figure 4 we show random function samples from a Gaussian process to illustrate the structure of the model. We have plotted samples for different length scales to demonstrate how they influence the variability of the GP.

2) Bayesian inference: In the preceding paragraph we have described the GP prior which encodes our beliefs about the output f of the computer code. The model is specified by parameters $\boldsymbol{\theta}$, which, for now, we assume to be fixed.

Assume we are given the results of N simulation runs $\mathcal{D} = \{(x_1, y_1) \dots (x_N, y_N)\}$, which are possibly corrupted by noise: $y_\ell = f(\mathbf{x}_\ell) + \epsilon_\ell$. As widely done, we assume that the noise is normal, i.e. $\epsilon \sim \mathcal{N}(\epsilon|0, \sigma^2)$. These data can now be used to update the model, to obtain what is called the posterior distribution.

The posterior process is obtained using Bayes' rule. It reflects our updated knowledge about f, including the remaining uncertainty. In the regression setup the posterior process is again a GP, and we can derive an analytic expression for the predictions at unseen inputs. At each input $\mathbf{x}^*$ the posterior

predictive distribution is normal

$$p(f^*(\mathbf{x}^*)|\mathcal{D}, \boldsymbol{\theta}) = \mathcal{N}(f^*|m(\mathbf{x}^*), v(\mathbf{x}^*)) \qquad (14\text{a})$$

$$\text{with mean} \quad m(\mathbf{x}^*) = \mathbf{k}(\mathbf{x}^*)^T Q^{-1} \mathbf{y} \qquad (14\text{b})$$

$$\text{and variance} \quad v(\mathbf{x}^*) = k(\mathbf{x}^*, \mathbf{x}^*) - \mathbf{k}(\mathbf{x}^*)^T Q^{-1} \mathbf{k}(\mathbf{x}^*) .$$

We have defined $Q = K + \text{diag}[\sigma^2, \ldots, \sigma^2]$, and used the abbreviations $\mathbf{k}(\mathbf{x}^*) \in \mathbb{R}^n$ and $K \in \mathbb{R}^{n \times n}$ with $[\mathbf{k}(\mathbf{x}^*)]_\ell = k(\mathbf{x}_\ell, \mathbf{x}^*)$ and $K_{i\ell} = k(\mathbf{x}_i, \mathbf{x}_\ell)$. The expressions (14) can be derived using basic algebra for multivariate Gaussian distributions, see [24, Chap. 2].

3) Learning the hyperparameters: In the above section we have introduced GPs with a given covariance function $k(\mathbf{x}, \bar{\mathbf{x}})$ and fixed noise level σ. The involved parameters $\boldsymbol{\theta}$ are so-called hyperparameters. In a correct Bayesian analysis we need to define an appropriate prior distribution $p_o(\boldsymbol{\theta})$ and average over their posterior distribution $p(\boldsymbol{\theta}|\mathcal{D})$:

$$p(f|\mathcal{D}) = \int \text{d}\boldsymbol{\theta} \quad p(\boldsymbol{\theta}|\mathcal{D}) \, p(f|\mathcal{D}, \boldsymbol{\theta}) . \qquad (15)$$

As the integral is not analytically tractable we have to resort to a numerical approximation.

One way to approximate the integral in (15) is to use Markov Chain Monte Carlo methods, as introduced by [14]. These methods give accurate approximations, but they are computationally very expensive and not easily handled. The maximum likelihood (ML) approximation of type two chooses the maximizer

$$\hat{\boldsymbol{\theta}} = \underset{\boldsymbol{\theta}}{\text{argmax}} \, [p(\mathcal{D}|\boldsymbol{\theta})] \qquad (16)$$

of the marginal likelihood. It is probably the most common approximation to replace the integral in (15). The method can be understood as shrinking the posterior distribution $p(\boldsymbol{\theta}|\mathcal{D})$ to a delta distribution at its mode $\hat{\boldsymbol{\theta}}$, thus using an overly confident estimate. The ML approach is justified for large sets of training data where $p(\mathcal{D}|\boldsymbol{\theta})$ is concentrated around its mode. MacKay [10] gives a detailed analysis. The marginal likelihood is given by a closed expression which can be optimized in a conjugate gradient scheme. Details can be found in [24, Chap. 5].

In comparison to correct averaging, choosing optimal parameters has the advantage that these can be used for screening. Welch et al. [32] describe an ML-type procedure which optimizes the length scale parameters and uses them to assess whether input parameters are inactive. This method is known as *Automatic Relevance Determination* (ARD), a term coined by MacKay [9]. The length scales $w_1 \ldots w_D$ are indirect parameters which have no meaningful physical interpretation. However, we can easily verify that for

$$w_d \gg \max_{j,k} \left| x_j^{(d)} - x_k^{(d)} \right| \quad \forall x_j, x_k \qquad (17)$$

the influence of the corresponding dimension becomes negligible in the covariance structure (13). The function is therefore constant along parameter $x^{(d)}$.

Although this screening approach can only roughly capture the functions' variability in the input dimensions, GPs can also be efficiently used to determine the statistical sensitivity measures: In the following section we derive analytic expressions for the variance based measures CR, LCR, and CCR.

4) Quadrature using GPs: As we have seen in Section III-A, Monte Carlo is the method of choice to replace classical methods for high dimensional quadrature. However, as O'Hagan [19] argues, MC methods have several drawbacks.

His main argument against MC methods is that only the function values $f(\mathbf{x}_\ell)$ enter the estimate, not the inputs $\mathbf{x}_\ell$ themselves. The Quasi-Monte Carlo methods, which we mentioned above, can to some extend relieve this disadvantage by using efficient sampling schemes. However, the very fact that the estimate (10) is unbiased implies an inefficient use of the available information. If we could incorporate prior knowledge about the function f we would introduce a bias, but could also use the available information more efficiently.

Gaussian process priors, which we introduced in Section III-B.1, introduce such prior information by assuming a concept of smoothness on f. If the prior assumptions do not prove wrong, we can expect an increase of efficiency in comparison to the MC methods which only assume square-integrability.

Rasmussen & Ghahramani [23] show how the Bayesian quadrature, as proposed in [20], can outperform classical MC. For Sensitivity Analysis Bayesian quadrature has been proposed by Haylock and O'Hagan [7], [16]. Besides a more efficient use of the data, the Bayesian approach has the advantage that the samples do not have to reflect the input distribution. We can therefore reduce the number of function evaluations by choosing an optimal design. One may also study the sensitivity's dependence on the nominal value $\hat{\mathbf{x}}$ after approximating the function in a larger region of the input space.

Having computed the posterior process $p(f|\mathcal{D}, \hat{\boldsymbol{\theta}})$, we can estimate mean or variance of the output under $p(\mathbf{x})$ (3) using the predictive mean $m(\mathbf{x})$ and variance $\sigma^2(\mathbf{x})$ (14b). The solution involves longish expressions, which we have relegated to the Appendix. The quadrature problem can be reduced to integrating products of the input distribution $p(\mathbf{x})$ and the covariance function. All necessary integrals are summarized by the following quantities:

$$k_c = \int \text{d}\mathbf{x} \, p(\mathbf{x}) \int \text{d}\mathbf{x}' \, p(\mathbf{x}') \quad k(\mathbf{x}, \mathbf{x}') \qquad (18)$$

$$k_o = \int \text{d}\mathbf{x} \, p(\mathbf{x}) \, k(\mathbf{x}, \mathbf{x})$$

$$z_\ell = \int \text{d}\mathbf{x} \, p(\mathbf{x}) \, k(\mathbf{x}, x_\ell)$$

$$L_{j\ell} = \int \text{d}\mathbf{x} \, p(\mathbf{x}) \quad k(\mathbf{x}, x_j) k(\mathbf{x}, x_\ell) .$$

Note that we can also calculate confidence intervals for the estimated quantities by taking into account the remaining uncertainty in the posterior process $p(f|\mathcal{D})$.

If we use the common covariance function (13), the integrals (18) can be calculated explicitly for some input distributions. For Gaussian input distributions

$$p(\mathbf{x}) = \mathcal{N}(\mathbf{x}|\hat{\mathbf{x}}, B) \qquad (19)$$

we give the solutions in the Appendix (see also [7]).

We can thus calculate mean and variance of $p_{\mathbf{x}}(f)$ in closed form, along with their gradients with respect to design
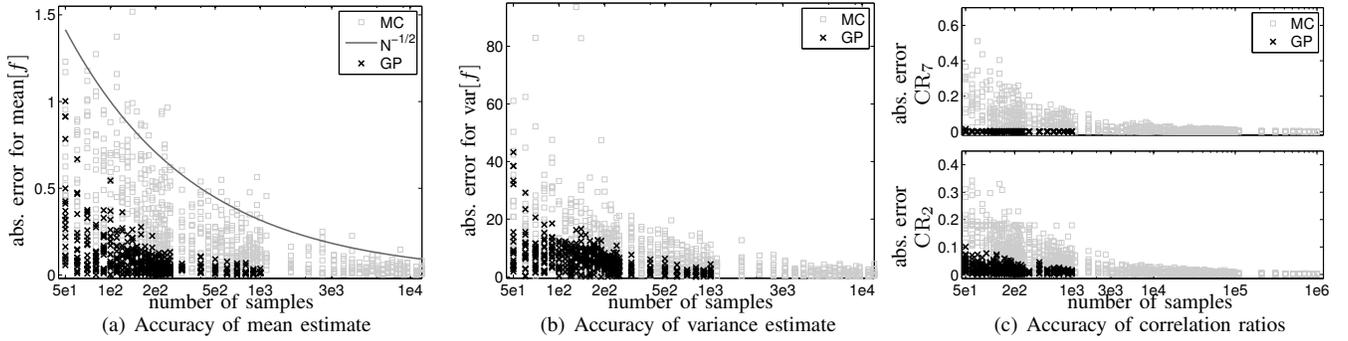
Fig. 5. Friedman's benchmark function: Convergence rates for the MC method and the GP-based Bayesian quadrature. Shown are the absolute error of the estimates for the mean (a), the variance (b) and two CR coefficients (c) against the number of samples. The true values can be calculated exactly and serve as a reference in all plots. The solid curve in the leftmost plot shows the characteristic error bound of the MC estimate for the mean, $\mathcal{O}(N^{-1/2})$.

parameters and input tolerances. Such gradients can be used for an efficient optimization to find optimally robust designs.

If the input distribution is factorized into independent factors $p(\mathbf{x}) = \prod_\ell p_\ell(x_\ell)$, the integrals in (18) break down to a product of one dimensional integrals. Those are, in contrast to a full integral of the type (3), relatively easy to handle. For most non-Gaussian input distributions the integrals cannot be calculated analytically, but can be approximated efficiently using the Gauss-Hermite rule for one dimensional quadrature.

The involved integrals can in general not be simplified and have to be solved numerically. This is the case for the parametric yield (4). However, having calculated a GP model based on the available data, we have access to a fast emulation of the original computer program. We can therefore calculate a simple Monte-Carlo estimate in the spirit of response surface methods, using a large number of samples [16].

The results of the above paragraph have been derived for a GP with zero mean, however, the generalization to nonzero mean functions is straightforward. Assume we add an offset $\mu(\mathbf{x})$ to the GP prediction $m(\mathbf{x})$ in (14). The expectations over $p(\mathbf{x})$ decompose to

$$\mathrm{mean}_\mathbf{x}\left[m(\mathbf{x}) + \mu(\mathbf{x})\right] = \mathrm{E}_\mathbf{x}\left[m(\mathbf{x})\right] + \mathrm{E}_\mathbf{x}\left[\mu(\mathbf{x})\right] \quad (20a)$$

$$\mathrm{var}_\mathbf{x}\left[m(\mathbf{x}) + \mu(\mathbf{x})\right] = \mathrm{var}_\mathbf{x}\left[m(\mathbf{x})\right] + \mathrm{var}_\mathbf{x}\left[\mu(\mathbf{x})\right] \quad (20b)$$
$$+ 2\mathrm{covar}_\mathbf{x}\left[m(\mathbf{x}), \mu(\mathbf{x})\right].$$

Thus, as long as we can compute the integrals over the products of $\mu(\mathbf{x})$, $p(\mathbf{x})$ and $k(\mathbf{x}, \cdot)$, it can easily be incorporated to the analysis. For a polynomial offset in combination with a Gaussian input distribution $p(\mathbf{x})$ all integrals are analytically tractable.

IV. EMPIRICAL STUDY OF ACCURACY AND EFFICIENCY

We have argued in Section III-B.4 that the Bayesian Monte Carlo analysis uses the available information efficiently by incorporating prior knowledge, and thus saves simulation effort. However, hereby we restrict the analysis to cases where the mapping f, given by the computer code, does not contradict these prior assumptions. To test the validity of the GP model we compare the accuracy of BMC and MC dependent on the number of simulation runs. We use real design problems and a well-known benchmark problem for nonlinear regression.

The re-usability of data, the separation of simulation and analysis, and the testability of the results via cross-validation are major advantages of the regression-based BMC method. Experimental design, which can only be applied in the Bayesian approach, can lead to an additional increase in efficiency [25]. In this section, however, we directly compare MC and Bayesian MC on identical designs from $p(\mathbf{x})$ to ensure a fair comparison.

A. Experiments

a) Analytical benchmark, Friedman's function: As computer models of our electro-mechanical systems are highly complex, the number of simulation runs which we can perform is limited. Therefore we additionally use an analytical benchmark function to compare the methods on a large number of samples.

We use a function which was defined by Friedman [6] as a benchmark problem for regression. It is nonlinear and non-monotonic, and therefore a challenging problem for Sensitivity Analysis. The function has 10 input parameters, 5 of them having an impact on the output. We use a normal input distribution $p(\mathbf{x}) = \mathcal{N}(\mathbf{x}|\hat{\mathbf{x}}, B)$ with mean $\hat{x} = (0, 0, \frac{1}{2}, 0 \ldots 0)$ and covariance $B = \mathrm{diag}(\frac{1}{4} \ldots \frac{1}{4})$. As we use a symmeterized version of the original function

$$f(\mathbf{x}) = 10\sin(\pi x_1 x_2) + 20(x_3 - \tfrac{1}{2})|x_3 - \tfrac{1}{2}| \quad (21)$$
$$+ 10x_4 + 5x_5,$$

we ensure that the mean under $p(\mathbf{x})$ is zero. As all sensitivity measures can be computed analytically we can compare the estimates to the true value. In this example we do not add noise, as this corresponds to the common situation in computer experiments. Note, however, that the BMC procedure handles noise automatically.

See Figure 5 for the convergence rates of the Bayesian Monte Carlo scheme in comparison to the classical MC method. The plots show the estimates for $\mathrm{mean}_\mathbf{x}[f]$, $\mathrm{var}_\mathbf{x}[f]$, and the CRs against the available number of function evaluations. The samples are drawn randomly, and thus the MC and BMC estimates are different for each repetition. We have plotted the results for a number of repetitions to illustrate the accuracy of the estimates.
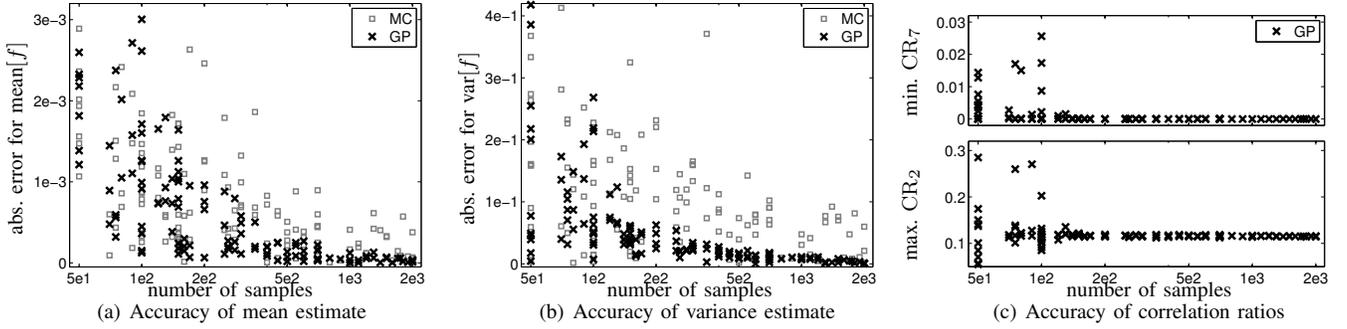
Fig. 6.   Performance of the BMC and MC methods on the PS model. Shown are the absolute error on the estimated mean (a) and variance (b), and the estimates for the minimal and maximal CR (c). As reference values we chose the mean of all estimates at $2 \cdot 10^3$ samples.

For both methods we have used identical Latin Hypercube designs, and therefore the comparison measures exclusively how efficient the available data is used. The BMC estimate of the mean and variance (6) using 10^3 samples is already as accurate as the MC estimate using 10^4 samples: This reduces the number of required simulation runs by a factor of ten. To calculate sensitivity measures like the CRs, the MC method requires an estimate for the variance and one additional MC run per parameter (7). However, even counting only the complexity of one MC run, BMC is still much more efficient. The accuracy of the MC approach on CR_2 with 10^5 samples is comparable to BMC using only 1% of the samples. CR_7—the input x_7 does not affect the output—is correctly identified to be zero on only 10^2 samples by the screening (ARD) capability of the GP, while the MC estimate only stabilizes around 10^5 samples.

b) The PS model: For the analytical example we have tested the methods on a large number of runs with up to 10^6 samples. Evidently, due to limited computational resources, we need to restrict the number of simulation runs on tests using the PS model. Find the results in Figure 6.

Our experiments show that BMC clearly outperforms the MC approach on this high dimensional sensor model. From the plots we can read that BMC estimates the mean as accurate as MC on $2\,000$ samples, using only 25% of the simulation runs. For the variance estimate this fraction is as low as 10%. The correlation ratios are apparently accurate within a few percent for the largest, as well as the for the smallest coefficient, using only 500 random samples. Note that we did not calculate the CRs using the MC method as this would have required an extra MC run per parameter.

c) Model 2, Accelerometer: Our second simulation code models the behavior of a micro electro-mechanical accelerometer which is used to trigger airbags or other automotive applications. The model has 29 parameters which show variations in the manufacturing process. The predictions of the GP model, trained on 300 points, lead to a root mean square error of 3.7% on an independent test set of $4\,700$ instances, relative to the test set's standard deviation.

It turns out that this model is dominated by linear effects, as indicated by the Sensitivity Analysis shown in Figure 7. The estimate of $\mathrm{var}_{\mathbf{x}}[f]$ converges much slower for the MC method than the BMC scheme: On 50 training instances BMC

is as accurate as MC on 500 samples. Both methods, however, perform comparably in estimating the $\mathrm{mean}_{\mathbf{x}}[f]$. This gap can be explained by the great linearity of the model. Due to linearity, all effects caused by a deviation from the nominal value cancel in the mean estimate, and effectively we only need to estimate the offset $f(\hat{\mathbf{x}})$. The MC method is here as efficient as the Bayesian approach. When we turn to the variation and the CRs, BMC can again profit from prior assumptions and shows extremely good estimates on only 50 samples.

B. Discussion

In the preceding empirical study we have compared Bayesian Monte Carlo to the traditional MC method for Sensitivity Analysis. We have focused on the accuracy which can be obtained using a limited number of simulation runs. As the implementation of the Monte Carlo method is much simpler than the proposed BMC scheme, it is certainly to be preferred when the models can be evaluated at low computational cost. When we consider complex models, however, the number of function evaluations is crucial.

We have used two computer models from the design process of MEMS and an analytical, nonlinear benchmark test to compare MC and BMC. The comparison on identical data sets showed that the BMC estimates require significantly less function evaluations than the MC method. This holds especially for the sensitivity measures and the output variation.

The increase in efficiency is based on the prior assumption of smooth behavior, which proves correct for the presented systems. In case of deflection from the prior assumption— e.g. for discontinuous functions—the BMC scheme cannot be applied. Such a case, however, can be detected when the meta-model is tested using e.g. cross validation.

V. SYNOPSIS

Fluctuations in the production process can have a substantial influence on the performance of integrated systems. A good design is characterized by its robustness with respect to these process-tolerances. Computer models can replace test runs in the design analysis, and can be used to study the characteristics of the system as design parameters are changed.

In high dimensional models one can no longer understand the influence of parameters by plotting projections to one

THIS IS A PREPRINT OF IEEE TRANS. ON SEMICONDUCTOR MANUFACTURING 19(4) PP. 475–486.

10

(a) Sensitivity Analysis for the accelerometer (b) Estimated mean and variation for the output (c) Estimates for minimal and maximal CR
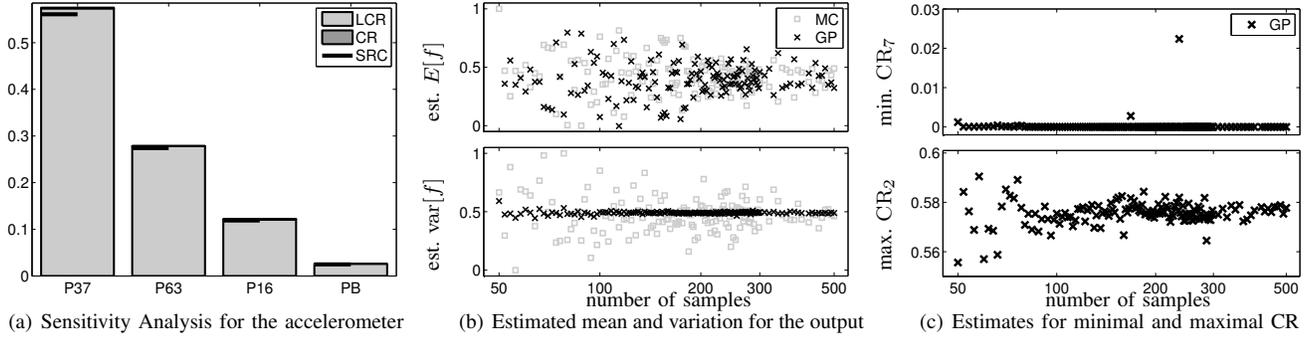
Fig. 7. Results for the accelerometer model. Shown are the Sensitivity Analysis (a), the estimated mean and variance (b), and the estimates for the minimal and maximal CR (c). The Sensitivity Analysis shows a linear behavior except in P36. It turns out that MC and BMC have similar convergence properties in the estimate of $\mathrm{mean}_\mathbf{x}[f]$, while the output variation $\mathrm{var}_\mathbf{x}[f]$ given by BMC converges much faster than the MC estimate. The CRs show little variation after roughly 300 samples.

or two axes. We define a number of statistically justified sensitivity measures to provide a compressed representation: In comparison they let the designer understand the impact of parameters, the degree of nonlinearity of the model, as well as the interaction of inputs.

Output fluctuations are traditionally assessed using Monte Carlo methods, where the parameters' distributions are emulated by a finite number of random samples. Monte Carlo is attractive by virtue of its simplicity and universal usability, however, it might not be feasible for computationally demanding models. We use a Bayesian approach based on nonparametric Gaussian process regression to make efficient use of available simulations runs. The sensitivity measures are calculated from the Gaussian process in closed form. The results can therefore be guaranteed to be as accurate as the tested regression model.

We have used two high-dimensional computer models of MEMS to demonstrate the effectiveness of the proposed method. In a case study we discuss the structure of one of the models to elucidate the proposed design analysis. We have thoroughly compared the convergence properties of our approach to MC. The analysis shows that the proposed BMC scheme can estimate the output variation as accurate as the MC method on only 10% of the number of simulation runs.

The BMC scheme separates the computer model from the design analysis, and—in contrast to the MC method—the inputs do not have do resemble the input distribution. We can therefore choose an optimal sampling scheme using experimental design. Also, we can scan a larger region of the input space instead of restricting the analysis to one setting of design parameters. Once the meta-model is trained, the designer can repeat the analysis for several settings nearly instantaneously, and the robustness of the design can be maximized automatically.

APPENDIX
MEAN AND VARIANCE FOR BAYESIAN MONTE CARLO

In the following paragraph we specify the estimates for the mean and variance of the output f under the input distribution $p(\mathbf{x})$ when a Gaussian process prior is used. The estimate for the mean is simply the average over the predictive mean, as

the expectations $\mathrm{E}_{f|\mathcal{D}}$ and $\mathrm{E}_\mathbf{x}$ can be swapped:

$$\mathrm{E}_{f|\mathcal{D}}\Big[\mathrm{E}_\mathbf{x}[f]\Big] = \mathrm{E}_\mathbf{x}\Big[m(\mathbf{x})\Big] = \int d\mathbf{x}\, p(\mathbf{x})\, m(\mathbf{x}) \quad (22a)$$
$$= \int d\mathbf{x}\, p(\mathbf{x})\, \mathbf{k}(\mathbf{x})^T Q^{-1}\mathbf{y} = \mathbf{z}^T Q^{-1}\mathbf{y}\,,$$

where we have used the definition of the mean $m(\mathbf{x})$ from (14b) and the abbreviation $\mathbf{z}$ from (18). The estimate of the variance implies also the predictive uncertainty, and we need to decompose it into three terms (see also [17]):

$$\mathrm{E}_{f|\mathcal{D}}\Big[\mathrm{var}_\mathbf{x}[f]\Big] = \mathrm{var}_\mathbf{x}\Big[\mathrm{E}_{f|\mathcal{D}}[f]\Big] + \mathrm{E}_\mathbf{x}\Big[\mathrm{var}_{f|\mathcal{D}}[f]\Big]$$
$$- \mathrm{var}_{f|\mathcal{D}}\Big[\mathrm{E}_\mathbf{x}[f]\Big]\,. \quad (22b)$$

If the GP model perfectly fits the function f and the predictive variance is zero, the sum reduces to the variance over the predictive mean:

$$\mathrm{var}_\mathbf{x}\Big[\mathrm{E}_{f|\mathcal{D}}[f]\Big] = \mathrm{var}_\mathbf{x}\Big[m(\mathbf{x})\Big] \quad (22c)$$
$$= \int d\mathbf{x}\, p(\mathbf{x})\,\big(\mathbf{k}(\mathbf{x})Q^{-1}\mathbf{y}\big)^2 - \mathrm{E}_\mathbf{x}[m(\mathbf{x})]^2$$
$$= \mathrm{trace}\big[(Q^{-1}\mathbf{y})(Q^{-1}\mathbf{y})^T L\big] - \mathrm{E}_\mathbf{x}[m(\mathbf{x})]^2\,,$$

where L is defined by (18). Due to finite predictive uncertainty we obtain the following two contributions:

$$\mathrm{E}_\mathbf{x}\Big[\mathrm{var}_{f|\mathcal{D}}[f]\Big] = \mathrm{E}_\mathbf{x}\Big[\sigma^2(\mathbf{x})\Big] \quad (22d)$$
$$= \int d\mathbf{x}\, p(\mathbf{x})\,\Big[k(\mathbf{x},\mathbf{x}) - \mathbf{k}(\mathbf{x})^T Q^{-1}\mathbf{k}(\mathbf{x})\Big]$$
$$= k_o - \mathrm{trace}[Q^{-1}L]]$$

$$\mathrm{var}_{f|\mathcal{D}}\Big[\mathrm{E}_\mathbf{x}[f]\Big] = \mathrm{E}_{f|\mathcal{D}}\Big[\big(\mathrm{E}_\mathbf{x}[f] - \mathrm{E}_{f|\mathcal{D}}[\mathrm{E}_\mathbf{x}[f]]\big)^2\Big] \quad (22e)$$
$$= \int d\mathbf{x}\, p(\mathbf{x})\int d\mathbf{x}'\, p(\mathbf{x}')\, \mathrm{E}_{f|\mathcal{D}}\Big[\big(f(\mathbf{x}) - \mathrm{E}_{f|\mathcal{D}}[f(\mathbf{x})]\big)$$
$$\times \big(f(\mathbf{x}') - \mathrm{E}_{f|\mathcal{D}}[f(\mathbf{x}')]\big)\Big]$$
$$= \int d\mathbf{x}\, p(\mathbf{x})\int d\mathbf{x}'\, p(\mathbf{x}')\, \mathrm{cov}_{f|\mathcal{D}}[f(\mathbf{x}), f(\mathbf{x}')]$$
$$= \int d\mathbf{x}\, p(\mathbf{x})\int d\mathbf{x}'\, p(\mathbf{x}')\,\big[k(\mathbf{x},\mathbf{x}') - \mathbf{k}(\mathbf{x})^T Q^{-1}\mathbf{k}(\mathbf{x}')\big]$$
$$= k_c - \mathbf{z}^T Q^{-1}\mathbf{z}\,,$$

where k_o and k_c are defined in (18). Note that the last term is identical to the predictive uncertainty for the mean-estimate.

When the common quadratic exponential covariance function (13) is used, the integrals can be solved in closed form for Gaussian input distributions $p(\mathbf{x})$ (19), and we obtain

$$
\begin{aligned}
k_o &= \int d\mathbf{x}\, p(\mathbf{x})\, k(\mathbf{x}, \mathbf{x}) = w_o^2 \quad\quad (23) \\
k_c &= \int d\mathbf{x}\, p(\mathbf{x}) \int d\mathbf{x}'\, p(\mathbf{x}')\, k(\mathbf{x}, \mathbf{x}') \\
&= w_o^2 \left| 2A^{-1}B + I \right|^{-\frac{1}{2}} \\
z_\ell &= \int d\mathbf{x}\, p(\mathbf{x})\, k(\mathbf{x}, \mathbf{x}_\ell) \\
&= w_o^2 (2\pi)^{\frac{d}{2}} |A|^{\frac{1}{2}} \int d\mathbf{x}\, \mathcal{N}(\mathbf{x}|\hat{\mathbf{x}}, B) \mathcal{N}(\mathbf{x}|\mathbf{x}_\ell, A) \\
&= w_o^2 |A^{-1}B + I|^{-\frac{1}{2}} \\
&\quad \times e^{-\frac{1}{2}\left[(\mathbf{x}_\ell - \hat{\mathbf{x}})^T (A+B)^{-1} (\mathbf{x}_\ell - \hat{\mathbf{x}}) \right]} \\
L_{j\ell} &= w_o^4 |2A^{-1}B + I|^{-\frac{1}{2}} \quad\quad (24) \\
&\quad \times e^{-\frac{1}{2}\left[(\mathbf{x}_j - \mathbf{x}_\ell)^T \frac{1}{2} A^{-1} (\mathbf{x}_j - \mathbf{x}_\ell) \right]} \\
&\quad \times e^{-\frac{1}{2}\left[(\bar{\mathbf{x}}_{\ell j} - \hat{\mathbf{x}})^T \left[\frac{1}{2} A + B \right]^{-1} (\bar{\mathbf{x}}_{\ell j} - \hat{\mathbf{x}}) \right]}
\end{aligned}
$$

with $\quad \bar{\mathbf{x}}_{\ell j} = \frac{1}{2} (\mathbf{x}_\ell + \mathbf{x}_j)$,

where we have defined $A = \mathrm{diag}\{w_1^2 \ldots w_D^2\}$. The mean and the covariance matrix of the input distribution are denoted by $\hat{\mathbf{x}}$ and B.

ACKNOWLEDGMENT

REFERENCES

[1] K. Antreich, H. E. Graeb, and C. U. Wieser, "Circuit analysis and optimization driven by worst-case distances," *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, vol. 13, no. 1, pp. 57–71, 1994.

[2] M. C. Bernardo, R. Buck, L. Liu, W. A. N. J. Sacks, and W. J. Welch, "Integrated circuit design optimization using a sequential strategy," *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, vol. 11, no. 3, pp. 361–372, 1992.

[3] G. E. Box and G. C. Tiao, *Bayesian Inference in Statistical Analysis*. Addison-Wesley, 1973.

[4] J. Classen, J. Franz, O. A. Prütz, and A. Kretschmann, "Design methodology for micromechanical sensors in automotive applications," in *Eurosensors*, 2004.

[5] S. W. Director, P. Feldmann, and K. Krishna, "Statistical integrated circuit design," *IEEE Journal of Solid-State Circuits*, vol. 28, no. 3, pp. 193–202, 1993.

[6] J. H. Friedman, "Multivariate adaptive regression splines," *The Annals of Statistics*, vol. 19, no. 1, pp. 1–67, 1991.

[7] R. G. Haylock and A. O'Hagan, "On Inference for Outputs of Computationally Expensive Algorithms with Uncertainty on the Inputs," in *Bayesian Statistics 5*. Oxford University Press, 1996, pp. 629–637.

[8] X. Li, J. Le, L. T. Pileggi, and A. Strojwas, "Projection-based performance modeling for inter/intra-die variations," *Proc. of the 2005 IEEE/ACM International conference on Computer-aided design*, pp. 721–727, 2005.

[9] D. J. C. MacKay, "Probable networks and plausible predictions - a review of practical bayesian methods for supervised neural networks," *Network: Computation in Neural Systems*, vol. 6, pp. 469–505, 1995.

[10] ——, "Comparison of approximate methods for handling hyperparameters," *Neural Computation*, vol. 11, no. 5, pp. 1035–1068, 1999.

[11] M. D. Mackay, R. J. Beckmann, and W. J. Conover, "A comparison of three methods for selecting values of input variables in the analysis of output from a computer code," *Technometrics*, vol. 21, no. 2, pp. 239–245, 1979.

[12] M. D. McKay, J. D. Morrison, and S. C. Upton, "Evaluating prediction uncertainty in simulation models," *Computer Physics Communications*, vol. 117, no. 1, pp. 44–51, 1999.

[13] R. H. Myers and D. C. Montgomery, *Response Surface Methodology: Process and Product Optimization Using Designed Experiments*. Wiley, 2002.

[14] R. Neal, "Probabilistic inference using markov chain monte carlo methods," University of Toronto, Tech. Rep. CRG–TR–93–1, 1993.

[15] H. Niederreiter, *Random number generation and quasi-Monte Carlo methods*. SIAM, 1992.

[16] J. E. Oakley and A. O'Hagan, "Bayesian Inference for the Uncertainty Distribution of Computer Model Outputs," *Biometrika*, vol. 89, pp. 769–784, 2002.

[17] ——, "Probabilistic sensitivity analysis of complex models: a Bayesian approach," *Journal of the Royal Statistical Society, Series B*, vol. 66, no. 3, pp. 751–769, 2004.

[18] A. O'Hagan, "Curve Fitting and Optimal Design for Prediction," *J.R. Statist. Soc. B*, vol. 40, no. 1, pp. 1–42, 1978.

[19] ——, "Monte Carlo is Fundamentally Unsound," *The Statistician*, vol. 36, no. 2/3, pp. 247–249, 1987.

[20] ——, "Bayes-Hermite Quadrature," *Journal of Statistical Planning and Inference*, vol. 29, no. 3, pp. 245–260, 1991.

[21] C. E. Rasmussen, "Matlab implementation of GP regression," 2001. [Online]. Available: www.gaussianprocess.org

[22] ——, "Gaussian processes to speed up hybrid monte carlo for expensive bayesian integrals," in *Bayesian Statistics 7*. Oxford University Press, 2003, pp. 651–659.

[23] C. E. Rasmussen and Z. Ghahramani, "Bayesian Monte Carlo," in *NIPS 15*. MIT Press, 2003, pp. 489–496.

[24] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning*. MIT Press, 2006.

[25] J. Sacks, S. B. Schiller, and W. J. Welch, "Design for computer experiments," *Technometrics*, vol. 31, no. 1, pp. 41–47, 1989.

[26] J. Sacks and D. Ylvisaker, "Design for regression problems with correlated errors," *The Annals of Mathematical Statistics*, vol. 37, no. 1, pp. 66–89, 1966.

[27] A. Saltelli, K. Chan, and E. M. Scott, Eds., *Sensitivity Analysis*. Wiley, 2000.

[28] A. Saltelli, S. Tarantola, and F. Campolongo, "Sensitivity analysis as an ingredient of modeling," *Statistical Science*, vol. 15, no. 4, pp. 377–395, 2000.

[29] S. S. Sapatnekar, P. Vaidya, and S.-M. Kang, "Convexity-based algorithms for design centering," *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, vol. 13, no. 12, pp. 1536–1549, 1994.

[30] I. Sobol, "Sensitivity estimates for nonlinear mathematical models," *Mathematical Modeling and Computational Experiment*, vol. 1, no. 4, pp. 407–411, 1993.

[31] H. P. Trah, J. Franz, and J. Marek, "Physics of semiconductor sensors," in *Advances in Solid State Physics 39*, vol. 39. Vieweg, 1999.

[32] W. J. Welch, R. J. Buck, J. S. Sacks, H. P. Wynn, T. J. Mitchell, and M. D. Morris, "Screening, prediction, and computer experiments," *Technometrics*, vol. 34, no. 1, pp. 15–25, 1992.