ECCOMAS Congress 2016
VII European Congress on Computational Methods in Applied Sciences and Engineering
M. Papadrakakis, V. Papadopoulos, G. Stefanou, V. Plevris (eds.)
Crete Island, Greece, 5–10 June 2016

COMPARISON BETWEEN A POLYNOMIAL CHAOS SURROGATE MODEL AND MARKOV CHAIN MONTE CARLO FOR INVERSE UNCERTAINTY QUANTIFICATION BASED ON AN ELECTRIC DRIVE TEST BENCH

Philipp Glaser[1], Michael Schick[1], Kosmas Petridis[1], and Vincent Heuveline[2]

[1]Robert Bosch GmbH
Robert-Bosch-Campus 1, 71272 Renningen, Germany
e-mail: {philipp.glaser, michael.schick3, kosmas.petridis}@de.bosch.com

[2] Engineering Mathematics and Computing Lab (EMCL)
University Heidelberg, 69117 Heidelberg, Germany
e-mail: vincent.heuveline@uni-heidelberg.de

Keywords: Polynomial Chaos Expansion, Sparse Grid, Bayes Inference, Electric Drive Test Bench, Markov Chain Monte Carlo

Abstract. *The development of uncertainty quantification schemes has been pushed forward due to the increasing demands for complex physical and computational simulation models. In industrial applications, distributions on model parameters play a crucial role and quantifing them is a big challenge.*

In this work, a test bench hardware is presented, which is designed to measure the motor characteristic of an electric drive. The special aspect of this setup is that parameter distributions, which in general are unknown, can be defined a priori. The obtained measurements serve as a reference to analyse the convergence of Polynomial Chaos (PC) and Markov Chain Monte Carlo (MCMC) in context of Bayesian inference.

Our focus is on analysing the feasibility of the PC approach as a surrogate model to replace the forward model in the Bayesian inference. In comparison to the classical approach, which directly uses the simulation model, we investigate the number of simulations needed to obtain a good estimation of the parameter distribution. In addition we use different orders for the PC expansion to fit the surrogate model. In our benchmark, we show that the PC expansion is able to significantly reduce the computational cost compared to a pure MCMC approach.

1 INTRODUCTION

Many industrial applications involve uncertain parameters, for which correct values are rarely available in a precise way and which have a significant impact on simulation results. Therefore, the quantification of such uncertainties plays a crucial role, for example, in face of unknown component tolerances or measurement errors. Among many others, one challenging task is to infer knowledge on parameter distributions in light of experimental data.

In this context, Bayesian inference [6, 7, 16] offers a framework to combine numerical simulations with observed experimental data in order to gain better knowledge on model parameter uncertainties. One popular method is the so-called Markov Chain Monte Carlo (MCMC) approach in which the posterior of the model parameters is updated using correlated Monte Carlo samples. However, in order to achieve a high numerical accuracy, usually many MCMC samples are required. This can become a large drawback, especially if the numerical cost of a single simulation becomes prohibitive. In contrast, surrogate models, like, for example, Polynomial Chaos (PC) expansions [9, 18] provide a meta description of model output variations with respect to the random input, making the exploration of the posterior parameter distribution numerically cheap. PC is a spectral approach which expands the stochastic solution in an orthogonal polynomial basis defined in terms of the input parameters. The corresponding coefficients in this expansion, which account for the major part of the numerical cost, may be determined by many different methods. In our work, we consider a projection based approach [11, 21] to compute the coefficients using a Smolyak sparse grid numerical cubature [15].

Furthermore, we built a test bench hardware, which is used to measure a motor characteristic of an electric drive with uncertain physical parameters. In our test bench hardware it is possible to define physical reference parameter distributions a priori and store an according set of measurements. This can be seen as a replacement of the electric drive with another one from series-production for every measurement. An alternative method to obtain those measurement data would be to pick out a large amount of electric drives from the production line. For each sample drive a motor characteristic has to be recorded under a standardized measurement setup. This would increase significantly the time and cost needed. Afterwards we use this reference data to compare the posterior distribution obtained from a PC surrogate with the posterior of the MCMC approach starting from various prior distributions, which are different to the distributions taken to generate our reference data. Our focus is on the analysis of the truncation error resulting from restricting the total polynomial degree of the PC expansion along with a comparison of the numerical effort for computing the PC coefficients to the number of samples required for a pure MCMC approach.

The remainder of this article is organized as follows. Section 2 introduces the used methods for the approximation of the posterior distributions. In addition the PC surrogate model is presented and a method for fitting this model is addressed. The summary of this part is given by a combination of the surrogate model with the Bayesian inference. In the following section 3 the benchmark problem is introduced. This includes on the one hand the test bench hardware and on the other hand the underlying simulation model. The identification and verification between those is one main challenge and analyzed in a subsection. Besides the simulation model in total, one focus is on the modeling of physical parameter at the input. This is based on the defined hyperpriors for the mean

and standard derivation which have to be mapped into one prior. The numerical results are presented in section 4. We conclude this work and provide an outlook on open research questions in Section 5.

2 UNCERTAINTY QUANTIFICATION

In the following, a model with a given uncertain parameter space is considered. For convenience the input and output connection of this forward problem can be represented by a deterministic mapping

$$\boldsymbol{y} = \mathcal{M}(\boldsymbol{x}),$$

where $\boldsymbol{x} = \{x_1, \ldots, x_N\}^T \in \mathbb{R}^N$ is a vector of the model parameters uncertainties. The vector $\boldsymbol{y} = \mathbb{R}^Q$ represents the model outputs. As the parameter vector $\boldsymbol{x}$ is assumed to be uncertain, we change notation from $\boldsymbol{x}$ to $\boldsymbol{X}$. The probability density function (PDF) $p_{\boldsymbol{X}}(\boldsymbol{x})$ is assumed to be with independent, scalar components x_i (finite noise assumption). Hence the model outputs are represented by a random vector, too, and are denoted as follows:

$$\boldsymbol{Y} = \mathcal{M}(\boldsymbol{X}).$$

2.1 BAYESIAN INFERENCE

The main issue of the Bayesian paradigm is to model uncertainties, e.g., arising from a lack of knowledge in a probabilistic way. In this work the focus is on model parameters $\boldsymbol{X}$ and any additional information about those can be described with a prior probability density $p_{\boldsymbol{X}}(\boldsymbol{x})$. In the absence of information, one may select a prior that is uninformative, e.g., a uniform distribution. Our goal is to update the knowledge about $\boldsymbol{X}$ by performing experiments and using the obtained observations $\boldsymbol{y}^{obs}$ to infer knowledge about the according pdf of $\boldsymbol{X}$. The connection between the model parameters and the gathered observations is expressed by the likelihood function $p(\boldsymbol{y}^{obs}|\boldsymbol{X})$, which requires the evaluation of a forward model. Using Bayes' rule, the posterior density for the model parameters can be written as

$$p(\boldsymbol{X}|\boldsymbol{y}^{obs}) = \frac{p(\boldsymbol{y}^{obs}|\boldsymbol{X})p(\boldsymbol{X})}{\int p(\boldsymbol{y}^{obs}|\boldsymbol{X})p(\boldsymbol{X})d\boldsymbol{X}}. \tag{1}$$

For convenience the likelihood function can be defined as $L(\boldsymbol{X}) := p(\boldsymbol{y}^{obs}|\boldsymbol{X})$, which may be viewed as a function of the model parameters $\boldsymbol{X}$. A classical approach is than to model the results from the forward problem with an additive noise:

$$\boldsymbol{y}^{obs} = \mathcal{M}(\boldsymbol{X}) + \boldsymbol{\epsilon},$$

where the components of $\boldsymbol{\epsilon}$ are independent and identically distributed random variables with density p_ϵ and the pdf of $\boldsymbol{\epsilon}$ is assumed to be normal distributed with mean $\mu = 0$ and variance $\sigma = \boldsymbol{\Sigma}$. In this simple case $\boldsymbol{\epsilon}$ includes both the model error and the measurement error, that can arise for example from sensor noise. The likelihood function can now be rewritten as follows:

$$L(\boldsymbol{X}) = \prod_i p_\epsilon(\boldsymbol{y}_i^{obs} - \mathcal{M}_i(\boldsymbol{X})), \tag{2}$$

where $\mathcal{M}_i(\boldsymbol{X}) := \mathcal{M}(\boldsymbol{X}^{(i)})$ denotes the model outputs at the i-th sample $\boldsymbol{X}^{(i)}$.The posterior distribution in equation (1) can be approximated by a Markov Chain Monte Carlo (MCMC) method. Therefore many simulation runs have to be evaluated, which often are expensive to compute due to the involved cost of evaluating the forward model $\mathcal{M}_i(\boldsymbol{X})$. Using a surrogate model, e.g. a Polynomial Chaos expansion, which is described in subsection 2.2, can provide a means to decrease the computational costs.

2.2 POLYNOMIAL CHAOS EXPANSION

The original polynomial chaos was first introduced by Wiener [18]. It uses the Hermite polynomials with respect to Gaussian random variables. It can also be extended to non-Gaussian measures [2]. Considering the given model $\mathcal{M}(\boldsymbol{X})$, the output is represented as a random vector $\boldsymbol{Y}$. Without loss of generality, a scalar output Y is assumed for simplicity in the following. The vector case can be obtained by applying the approach componentwise to the model outputs using the same polynomial basis for each component.

2.2.1 DEFINITION IN ONE-DIMENSIONAL CASE

Assume the probability space $(\Theta, 2^\Theta, \mathbb{P})$, where Θ is the sample space and $\theta \in \Theta$ is a sample of it. 2^Θ is the σ-algebra and $\mathbb{P}$ is some probability measure. Now, let $Y(\theta)$ be a random process, treated as a function of θ. The spectral stochastic representation considered in this work can be seen as an Fourier-like decomposition [12]:

$$Y(\theta) = \sum_i y_i \phi_i(\xi(\theta)),$$

where y_i are the spectral coefficients, which have to be determined. The hyperparameter ξ is defined on the support Θ, corresponding to type of basis polynomials. $\{\psi_i\}_{i=0}^{\infty}$ denote the basis functions, which form an orthogonal basis with the following orthogonality relation:

$$\langle \phi_j(\xi(\theta)), \phi_k(\xi(\theta)) \rangle = \mathbb{E}\left[\phi_j(\xi(\theta))\phi_k(\xi(\theta))\right] = \mathbb{E}\left[\phi_k^2(\xi(\theta))\right]\delta_{jk}, \quad j,k \in \mathbb{N}_0, \tag{3}$$

with the Kronecker delta δ_{jk} which is equal 1 if $j = k$ and 0 otherwise. $\mathbb{E}\left[\phi_k^2(\xi(\theta))\right]$ is a normalization factor. To simplify the notation, we shall drop the dependence of the hyperparameter ξ on θ. Suppose that the cumulative distribution function $F(\xi)$ of Y is absolutely continuous w.r.t. the Lebesque-measure, then the corresponding PDF exists, such that $dF(\xi) = \rho(\xi)d\xi$. The inner product from equation (3) can be written as:

$$\langle \phi_j(\xi), \phi_k(\xi) \rangle = \mathbb{E}\left[\phi_j(\xi)\phi_k(\xi)\right] = \int \phi_j(\xi)\phi_k(\xi)\rho(\xi)d\xi, \quad j,k \in \mathbb{N}_0.$$

Some examples on the correspondence between common continuous distributions on Y and the type of the generalized PC basis polynomials and the support Θ for the hyperparameter ξ [21] can be found in Table 1.

2.2.2 DEFINITION IN MULTI-DIMENSIONAL CASE

Now we assume the hyperparameter ξ to be a random vector $\boldsymbol{\xi} = \{\xi_1, \ldots, \xi_N\}$ with mutually independent components. The orthogonality relation from equation (3) immediately takes the form (due to independence):

$$\langle \phi_j(\boldsymbol{\xi}), \phi_k(\boldsymbol{\xi}) \rangle = \mathbb{E}\left[\phi_j(\boldsymbol{\xi})\phi_k(\boldsymbol{\xi})\right] = \int \phi_j(\boldsymbol{\xi})\phi_k(\boldsymbol{\xi})\boldsymbol{\rho}(\boldsymbol{\xi})d\boldsymbol{\xi}, \quad 0 \leq j,k \leq \infty, \tag{4}$$

Table 1: Correspondence between the type of generalized PC polynomial basis and probability distribution.

distribution	generalized PC basis polynomials	support
Beta	Jacobi	$[a, b]$
Gamma	Laguerre	$[0, \infty]$
Gaussian	Hermite	$(-\infty, \infty)$
Uniform	Legendre	$[a, b]$

where $\boldsymbol{\rho}(\boldsymbol{\xi}) = \prod_{i=1}^{N} \rho_i(\xi_i)$. Let $\boldsymbol{i} = \{i_1, \ldots, i_N\} \in \mathbb{N}_0^N$ denote a multi-index with $|\boldsymbol{i}| = i_1 + \cdots + i_N$. The N-variate generalized PC basis functions can be obtained by the product of the univariate generalized PC polynomials in equation (4):

$$\psi_{\boldsymbol{i}}(\boldsymbol{\xi}) = \prod_{j=1}^{N} \phi_{i_j}(\xi_j).$$

Then, the generalized PC expansion reads:

$$Y(\boldsymbol{\xi}) = \sum_{|\boldsymbol{i}|=0}^{\infty} y_{\boldsymbol{i}} \psi_{\boldsymbol{i}}(\boldsymbol{\xi}),$$

where $y_{\boldsymbol{i}}$ are the unknown PC coefficients. In addition the multi-index $\boldsymbol{i}$ can be mapped into a one dimensional index by a one-to-one correspondence, which results in the following relation:

$$Y = \sum_{j=0}^{\infty} y_j \psi_j(\boldsymbol{\xi}). \tag{5}$$

For computational reasons the given polynomials have to be truncated at order p. The approximation is denoted as $\tilde{Y}$ and can be written as:

$$\tilde{Y} = \sum_{j=0}^{P} \tilde{y}_j \psi_j(\boldsymbol{\xi}), \quad \text{with } P + 1 = \frac{(N+p)!}{N!p!}. \tag{6}$$

In a second step the PC coefficients $\tilde{y}_j$ for $j = 0, \ldots, P$ in equation (6) have to determined. The PC coefficients can be recast each as a multidimensional integral which can be computed by multivariate numerical quadrature rule. This approach is called the projection method and will be used in the following.

Starting from the expansion of the solution in equation (6) the projection method projects the solution $\tilde{Y}$ against each basis function $\{\psi_i\}$ using inner product:

$$\left\langle \tilde{Y}, \psi_i(\boldsymbol{\xi}) \right\rangle = \left\langle \sum_{j=0}^{P} \tilde{y}_j \psi_j(\boldsymbol{\xi}), \psi_i(\boldsymbol{\xi}) \right\rangle. \tag{7}$$

By applying the orthogonality property of the basis polynomials in equation (3), the

right-hand side of the equation (7) can be modified as follows:

$$\begin{aligned}\mathbb{E}\left[\tilde{Y}\psi_j(\boldsymbol{\xi})\right] = \left\langle \tilde{Y}, \psi_i(\boldsymbol{\xi})\right\rangle &= \left\langle \sum_{j=0}^{P} \tilde{y}_j \psi_j(\boldsymbol{\xi}), \psi_i(\boldsymbol{\xi})\right\rangle \\ &= \sum_{j=0}^{P} \tilde{y}_j \mathbb{E}\left[\psi_j(\boldsymbol{\xi})\psi_i(\boldsymbol{\xi})\right] \\ &= \sum_{j=0}^{P} \tilde{y}_j \mathbb{E}\left[\psi_j^2(\boldsymbol{\xi})\right] \delta_{ij}.\end{aligned}$$

Thereby the PC coefficients result in

$$\tilde{y}_j = \frac{1}{\mathbb{E}\left[\psi_j^2(\boldsymbol{\xi})\right]} \mathbb{E}\left[\tilde{Y}\psi_j(\boldsymbol{\xi})\right], \quad \text{for } j = 0, \ldots, P,$$

with $\tilde{Y}$, that can be seen as an approximation of the random process. If we assume that $\boldsymbol{\xi}$ is continuous then the orthogonality can be written as:

$$\tilde{y}_j = \frac{1}{\mathbb{E}\left[\psi_j^2(\boldsymbol{\xi})\right]} \int_{\mathbb{R}^N} \tilde{\mathcal{M}}(\boldsymbol{\xi})\psi_j(\boldsymbol{\xi})\rho(\boldsymbol{\xi})d\boldsymbol{\xi}, \quad \text{for } j = 0, \ldots, P, \tag{8}$$

where $\rho(\boldsymbol{x})$ is the multivariate distribution density. The PC coefficients in equation (8) can be estimated numerically based on discretization. The integration can be done e.g. using a full tensor quadrature rule with order r and weights w_i. Hence the total number of quadrature points computes to:

$$R = (r)^N \tag{9}$$

and the approximation of the PC coefficients can be expressed as:

$$\tilde{y}_j \approx \hat{y}_j = \frac{1}{\mathbb{E}\left[\psi_j^2(\boldsymbol{\xi})\right]} \sum_{i=1}^{R} \tilde{Y}(\xi_i)\psi_j(\xi^{(i)})w_i, \quad \text{for } j = 0, \ldots, P. \tag{10}$$

2.3 SPARSE GRID

One limitation of the full grid approach is the severe exponential growth of the required simulations related to the number of dimensions in the parameter space. Given that, modifications has to be done to make the method applicable to complex models. One approach considers the choice of the nodes where the model is evaluated by selecting 'important' points. This is called sparse grid and it can reduce the number of evaluations appreciably.

Based on a spectral projection method described in the previous section 2.2 it is essential to give a selection of nodes to approximate the multidimensional integral in equation (8) with a discrete sum given in equation (10). We use a set of points generated by Gauss quadrature. This rule approximates best for the one-dimensional case ($N = 1$) [19, 21]. In the multi-dimensional case ($N > 1$) the total number of simulation points grows fast in high dimensional random parameter space.

It is obvious from equation (9) that even for a proportionally small parameter space $N \gg 1$ and an adequate PC order of $p = 3$, it can be assumed that $Q = 4^N \gg 1$. With

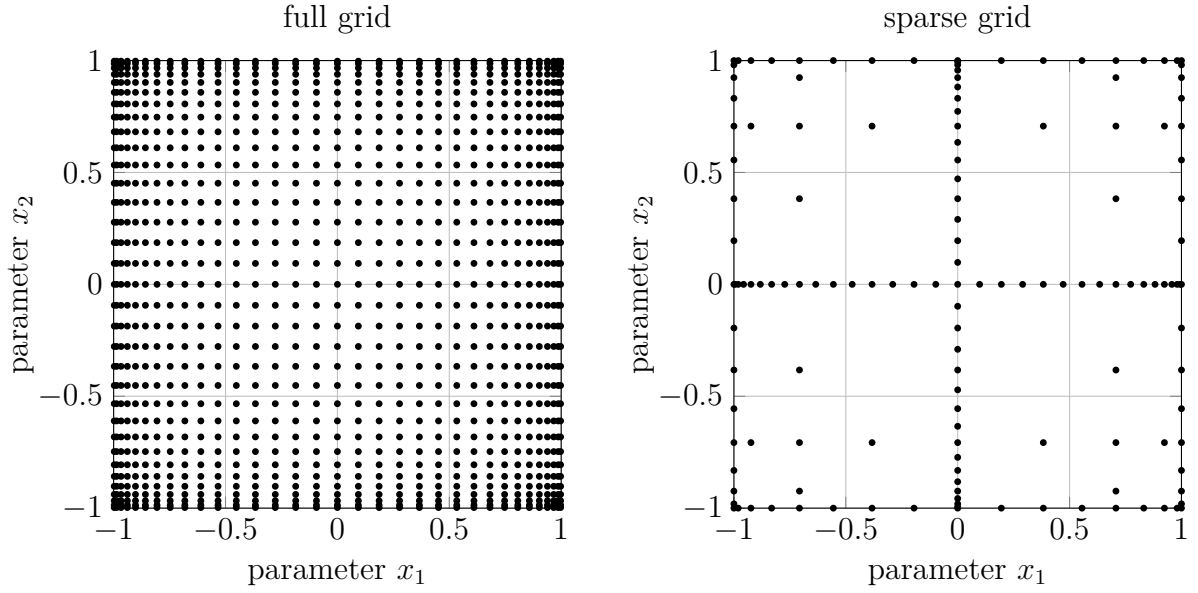

Figure 1: Comparison of two-dimensional grids based on the same one-dimensional nodes based on a two-dimensional parameter space. Left: full grid with 1089 nodes. Right: sparse grid with 145 nodes.

respect to this property the full grid rule is only used at lower dimensions. To reuse the spectral projection method sparse grids are proposed in this context [20]. Those were introduced in [15] and they are first used in the context of multivariate integration [8, 14].

The Smolyak algorithm is a linear combination of product formulas and based on a one-dimensional interpolation formula for an univariate function f

$$Q_j^1 f := \sum_{i=1}^{n_j^1} w_{ji} f(x_{ji}),$$

with the nodal sets $\Theta_j^1 = \left(x_{j1}, \ldots, x_{jn_j^1}\right)$. The Smolyak algorithm [8] is given:

$$Q_j^k f = \sum_{j \leq |\boldsymbol{i}|_1 \leq j+k-1} (-1)^{j+k-|\boldsymbol{i}|_1-1} \begin{pmatrix} k-1 \\ |\boldsymbol{i}|_1 - j \end{pmatrix} (Q_{i_1}^1 \otimes \cdots \otimes Q_{i_k}^1) f, \tag{11}$$

where $\boldsymbol{i} = (i_1, \ldots, i_N) \in \mathbb{N}$. The points of the multivariate Smolyak formula in equation (11) form a sparse grid given by the union over the parwise disjoint nodal sets:

$$\Theta_N = \bigcup_{j \leq |\boldsymbol{i}|_1 \leq j+k-1} (\Theta_{i_1}^1 \otimes \cdots \otimes \Theta_{i_k}^1).$$

Figure 1 shows the comparison of a full and sparse grid based on a two dimensional parameter space. It can be seen that the sparse grid based on the Smolyak algorithm [15] is a subset of the full grid and it consists of less simulation points than the full tensor grid. This opens up the possibility of considering higher dimensional random parameter spaces in combination with the PC expansion [1, 5, 11].

2.4 BAYESIAN INFERENCE WITH A PC SURROGATE MODEL

The integration of the PC surrogate model with a suitable sparse grid approach and the Bayesian inference can be done in the following way. The assumptions on the parameter

distributions can be used to define priors $p(\boldsymbol{X})$ for the uncertain model parameters. With knowledge of the PC coefficients obtained in equation (10), an approximation for the density function of model output $\hat{Y} = \hat{\mathcal{M}}_p(\boldsymbol{\xi}(\boldsymbol{X}))$ with order P, which can be seen as an surrogate model.To simplify the notation, we shall drop the dependence of the hyperparameter $\boldsymbol{\xi}$ on $\boldsymbol{X}$ and the PC surrogate model reads:

$$\hat{\mathcal{M}}_P(\boldsymbol{\xi}) = \sum_{j=0}^{P} \hat{y}_j \psi_j(\boldsymbol{\xi}).$$

The likelihood function $L(\boldsymbol{X})$ from equation (2) results in the following approximation:

$$\hat{L}(\boldsymbol{X}) = \prod_i p_\epsilon \left(y_i^{obs} - \hat{\mathcal{M}}_P(\boldsymbol{\xi}) \right) = \prod_i p_\epsilon \left(y_i^{obs} - \sum_{j=0}^{P} \hat{y}_j \psi_j(\boldsymbol{\xi}) \right). \tag{12}$$

Applying the obtained result to equation (1) and excluding the normalization in the denominator for an instant, one achieve the following:

$$p(\boldsymbol{X}|\boldsymbol{y}^{obs}) \propto L(\boldsymbol{X}) p(\boldsymbol{X}) = \prod_i p_\epsilon \left(y_i^{obs} - \sum_{j=0}^{P} \hat{y}_j \psi_j(\boldsymbol{\xi}) \right) p(\boldsymbol{X}). \tag{13}$$

As mentioned above, one approach to estimate the posterior parameter distributions $p(\boldsymbol{X}|\boldsymbol{y}^{obs})$ is to employ MCMC methods, which construct a Markov chain on the parameter space $\boldsymbol{X}$, whose steady state distribution is the posterior distribution of interest.

3 BENCHMARK PROBLEM

One main challenge is the validation of the resulting posterior distributions by applying the methods mentioned in section 2 to a real application case. One reason is that parameter distributions can not easy obtained in many cases or determining them is very expensive. On the other hand measurements of the outputs can be gathered simply.

For this purpose a test bench hardware is built up and there it is possible to vary the voltage supply and the winding resistance in an automatic way. This offers the possibility to define parameter distributions and run the test bench on them. For the resulting measurements the based parameter variations are known. Besides that, different motor characteristics can be chosen and recorded. The given model of an electric drive consists of a motor and a worm gear. Besides the electrical and mechanical relations, a detailed thermal model is available. To obtain a good fitting between the test bench and the model, a parameter identification, described in section 3.3, was performed.

3.1 TEST BENCH HARDWARE

The electric drive (cf. figure 2, ①) contains the motor and the gearing and is attached to a plate, which is connected to the mounting (cf. figure 2, ⑥) of the test bench. Several temperature sensors are fixed to the motor and the gear housing. The motor shaft is connected with a torque sensor (cf. figure 2, ②). In addition the rotational speed can be measured. The electromagnetic powder break (cf. figure 2, ③) enables the test bench to perform different series of measurements, which can range from static operating mode through motor characteristics. To compensate the resulting shaft misalignments, metal bellow-type couplings (cf. figure 2, ⑤) are used for the link between the components. The electric drive is connected with a power supply (cf. figure 2, ④).

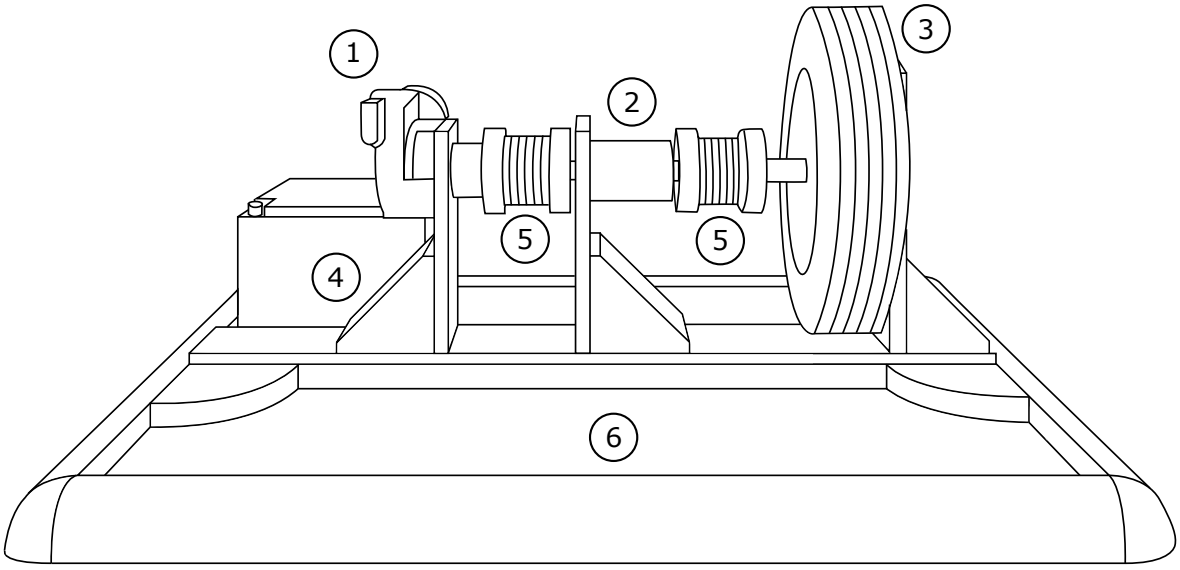

Figure 2: Schematic view on the test bench hardware.

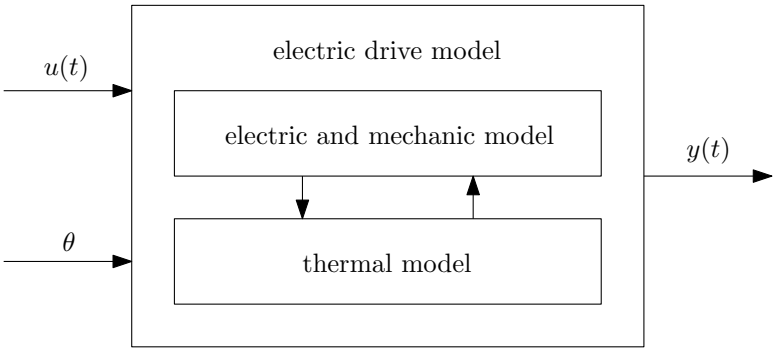

Figure 3: Schematic view on the electric drive model with the input/output connections.

The special feature of the test bench is the possibility to add uncertainties, which influences the power supply voltage of the drive and the characteristics of the electric motor. Hence, this can be done in an automatic way without replacing the electric drive or the power supply. For this reason many measurements based on given parameter variations can be generated.

3.2 SIMULATION MODEL

The given model of an electric drive consists of a motor and a worm gear. Beside the electrical and mechanical relations, a detailed thermal model is provided. They are strongly connected in both directions by internal states, cf. Fig. 3. The electric drive has eight model parameters and one input which is considered as uncertain. We assume that it is independent and identically distributed with a Gaussian distribution. To get a general view on the model, simplified equations are representing the correlations of the parts. The electric and mechanic part is characterized by the two states which are the current $I(t)$ and motor speed $\omega(t)$. The ordinary differential equations are given as

follows:

$$\frac{d}{dt}I(t) = \frac{1}{L}\Big[- RI(t) - c_m\omega(t) + \Big(U(t) - U_{drop}(I(t))\Big)\Big], \tag{14}$$

$$\frac{d}{dt}\omega(t) = \frac{1}{J}\Big[- \tau_{load}(t)i_g + \Big(\Big(\tau_{motor}(I(t), c_m)\Big) - \tau_{fric}\Big(\omega(t), \eta\Big)\Big)\eta\Big]. \tag{15}$$

The inputs of the simulation model are the voltage $U(t)$ and the load torque $\tau_{load}(t)$ which are summarized as $\boldsymbol{u}(t)$. The parameters are defined as the winding resistance R, the motor constant c_m, the inductance L, the gear ratio i_g and the gear efficiency η. The voltage drop $U_{drop}(I(t))$ is approximated by a polynomial of high order depending on the current $I(t)$.

The motor torque τ_{motor} can be characterized by the following equation:

$$\tau_{motor}(I(t), c_m) = c_m I - \tau_{loss}(\epsilon_{iron}),$$

where the losses in the copper, the iron and by hysteresis are summarized. They can be adapted by the parameter ϵ_{iron}. The mechanical friction losses can be described by the following nonlinear function:

$$\tau_{fric} = f(\omega(t), \eta, \mu_A, \mu_B),$$

which only shows the dependences for the sake of simplicity. The losses in the bearings A and B can be modified by the parameters μ_A and μ_B.

The thermal model of the electric drive is subdivided into several parts related to the mechanical structure. The link with the two states given in equations (14) and (15) are provided with the thermal power losses. The additional states are the temperatures at

- the magnet T_{magnet},
- the bearings A $T_{bearingA}$ and B $T_{bearingB}$,
- the minus brush T_{brush},
- the worm T_{worm} and
- the coil T_{coil}.

The interconnection with the electrical and mechanical model are as follows:

$$\begin{aligned}
R &= R_0\left[1 + \alpha\left(T_{coil} - T_{amb}\right)\right], \\
c_m &= \sum_{n=0}^{6} k_p(n) I^n, \quad \text{with } k_p(n) = \sum_{i=0}^{6} b_n(i) T_{magnet}^i, \\
\eta &= g(T_{worm}), \quad g \in \text{nonlinear function}, \\
\tau_{fric} &= h(T_{beraingA}, T_{beraingB}) \quad h \in \text{nonlinear function}.
\end{aligned}$$

3.3 OBTAINING MEASUREMENT DATA AND PARAMETER IDENTIFICATION

On main part of the benchmark problem is the parameter identification of the introduced model in subsection 3.2 with the measurements obtained from the test bench mentioned in subsection 3.1. In this paper, the temperatures of the system supposed to be at a stationary level and therefore the measurements have to be taken in a bounded temperature range:

$$T_{brush} = T_{brush,stat} \pm 1[K], \quad T_{magnet} = T_{magnet,stat} \pm 0.5[K],$$
$$T_{winding} = T_{winding,stat} \pm 0.5[K], \quad T_{worm} = T_{worm,stat} \pm 1.5[K].$$

This can be obtained by monitoring the temperature levels and run a warm-up sequence before the measurements are recorded. A measurement i starts at time t_0^i and records the time responses of the outputs. After a fixed time Δt the measurement is finished. The resulting data can be called measurement run i and contains $I_{measured,i}(t)$, $\omega_{measured,i}(t)$ within the range $t \in [t_0^i, t_0^i + \Delta t]$. Moreover, the voltage $U(t)$ is set to a stationary value in run i using the Heaviside step function:

$$U(t) = \hat{U}_i H(t), \quad t \in [t_0^i, t_0^i + \Delta t].$$

Before the next run is started, the electric drive is switched off by setting the voltage $U(t > t_0^i + \Delta t)$ to zero. The next run $i+1$ will then start at time $t_0^{i+1} > t_0^i + \Delta t$ with stationary voltage $\hat{U}_{i+1}$. This offers the opportunity to vary the model input $\hat{\boldsymbol{U}} = [\hat{U}_1, \hat{U}_2, \dots]$ over the measurement runs $1, 2, \dots$, or to put it another way, the electric drive is switched on in every run and the variation can be seen in the different stationary input voltages.

For the identification of the simulation model a measurement run at $\hat{U}_{ident}$ is performed and the obtained data over time is used to fit the physical model parameters:

$$\theta = [R, c_m, L, J, \eta_{iron}, \mu_A, \mu_B, \mu_{worm}] .$$

This nonlinear optimization problem is solved using the Matlab Optimization Toolbox [17] with the function lsqnonlin which includes a large-scale algorithm. This algorithm is a subspace trust-region method and is based on the interior-reflective Newton method described in [3] and [4]. Each iteration involves the approximate solution of a large linear system using the method of preconditioned conjugate gradients. The estimated parameters are denoted as $\hat{\theta}$ in the following.

3.4 MODELING OF PHYSICAL PARAMETER

Instead of using a non-parametric Bayesian model, it is assumed that the distribution of the physical parameter U belongs to a Normal distribution. This leads to a parametric model which has to be extended in the following way:

$$\tilde{U}(\mu_U, \sigma_U, \xi_U) = \mu_U + \sigma_U \xi_U,$$

with the hyperprior μ_U and the hyperprior σ_U. The parameter ξ_U is Normal distributed with expected value 0 and standard derivation 1. The interconnection of the parameter and the simulation model is shown in figure 4.

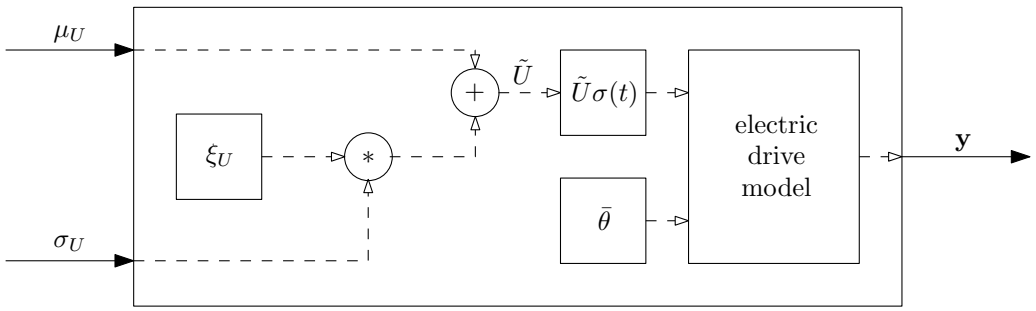

Figure 4: Parameter mapping.

Using the parametric model the likelihood function in equation (13) can be rewritten in the following way:

$$\hat{L}(\boldsymbol{X}) = \prod_i p_\epsilon \left(y_i^{obs} - \sum_{j=0}^{M} \hat{y}_{i,j} \psi(\xi_\mu(\mu_U), \xi_\sigma(\sigma_U), \xi_U) \right),$$

where $\xi_\mu(\mu_U)$ and $\xi_\sigma(\sigma_U)$ are hyperparameters of the PC surrogate model with the mapping:

$$\xi_\mu(\mu_U) = \frac{\mu_U - E[P(\mu_U)]}{\sqrt{Var(P(\mu_U))}}, \quad \xi_\sigma(\sigma_U) = \frac{\sigma_U - E[P(\sigma_U)]}{\sqrt{Var(P(\sigma_U))}}.$$

4 NUMERICAL RESULTS

We assume a constant load scenario with $\tau_{load} = 3Nm$ and a stationary input voltage $\hat{U}$. For this reason the observed motor speed ω will reach a steady state. In the following all given quantities are normalized with a fixed maximum value.

The physical parameter of the stationary voltage $\hat{U}$ is assumed to be Normal distributed with the expected value $\mu_{U,real} = 0.9$ and the standard derivation $\sigma_{U,real} = 0.047$. The number of measurement runs is set to 100 and Δt is fixed to 10 [s]. The data can be obtained using $\hat{\boldsymbol{U}} = [\hat{U}_1, \ldots, \hat{U}_{100}]$, which are drawn from the assumed distribution. Based on the setting a steady state of the motor speed $\omega_{measured,i}(t)$ will be reached after time $t = t_{steady}$ with $0 < t_{steady} \leq \Delta t$ for each run i and a stationary value for the motor speed $\omega_{steady,i} = \omega_{measured,i}(t_{steady})$ can be obtained. This means that for every measurement run a scalar value for the motor speed is taken into account, instead of a time behavior.

Based on the measurement runs $i = 1, \ldots, 100$ an error ϵ_i between the data $\omega_{meas,i} = \omega_{steady,i}$ and the simulation model $\omega_{sim,i} = \mathcal{M}(\hat{U}_i, \hat{\theta})$ can be defined. This is illustrated in figure 5. The resulting error ϵ can be described by the stochastic moments:

$$\mu_\epsilon = 0.00015, \quad \sigma_\epsilon = 0.00603.$$

Applying the MCMC method to the described benchmark problem and the measured data, the Metropolis-Hastings algorithm [10, 13] is used for the estimation of posterior distribution. The priors are assumed to be Uniform distributed with the following stochastic moments:

$$P(\mu) = \mathcal{U}(0.65, 1), \quad P(\sigma) = \mathcal{U}(0, 0.065).$$

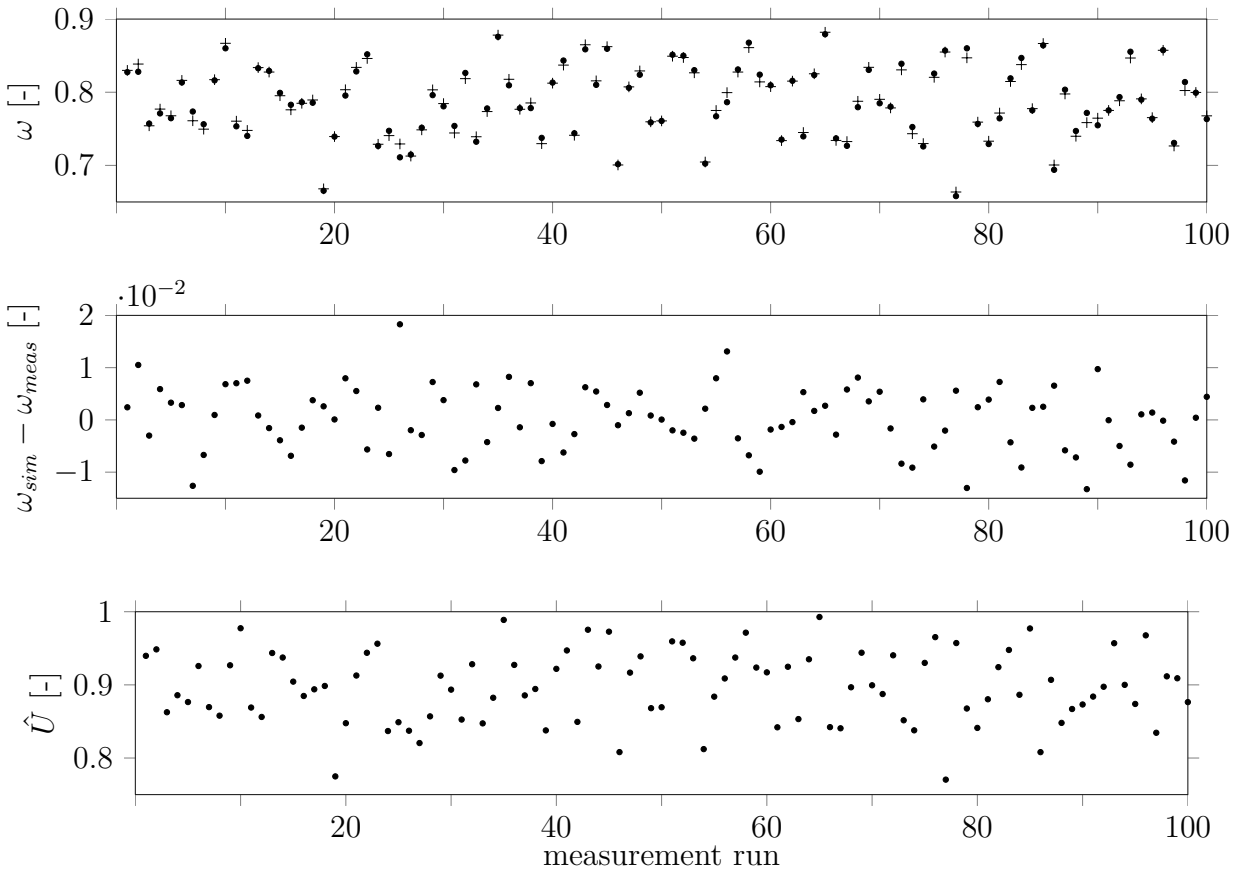

Figure 5: The first plot shows the normalized model output ω at a stationary time point t_{steady} for the measurement ($\bullet$) and the simulation result ($+$). The second plot shows the error ϵ between the measurement and the simulation output. The third plot shows the corresponding normalized voltage for the model evaluation.

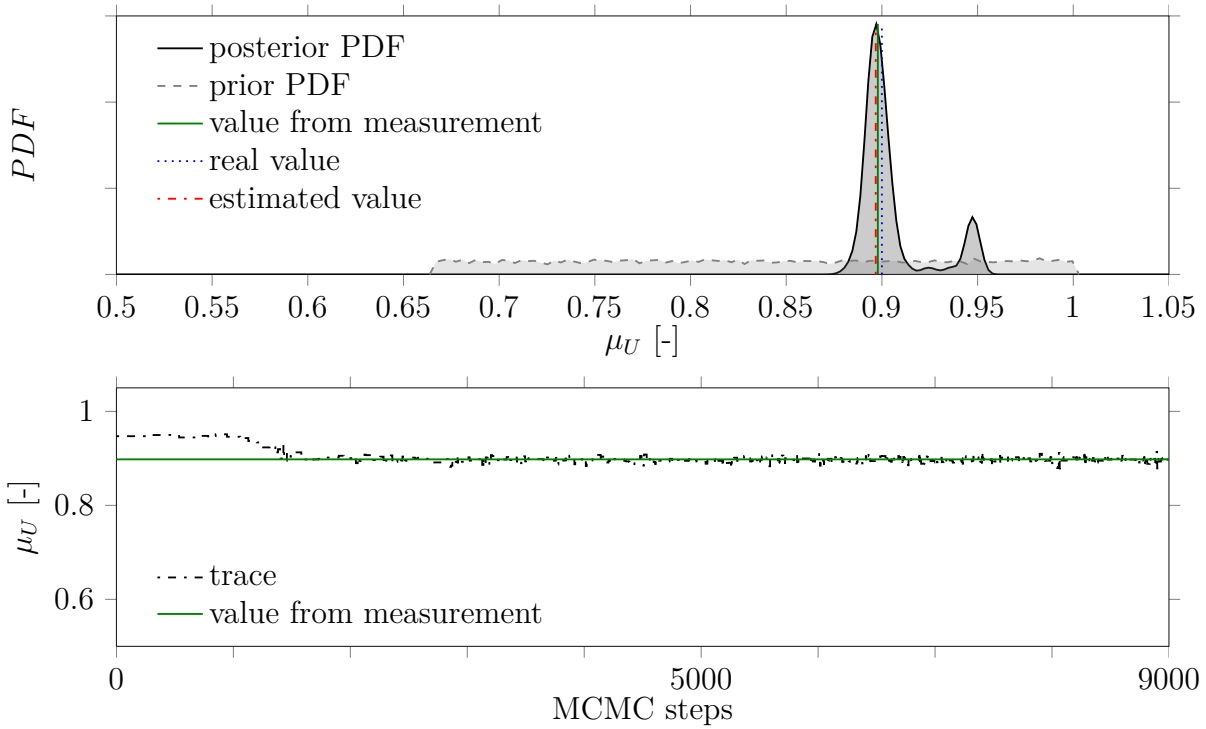

Figure 6: This figure shows the result of the MCMC estimation. The first plot contains the prior and posterior PDF of the hyperparameter μ_U. The second plot visualize the trace.

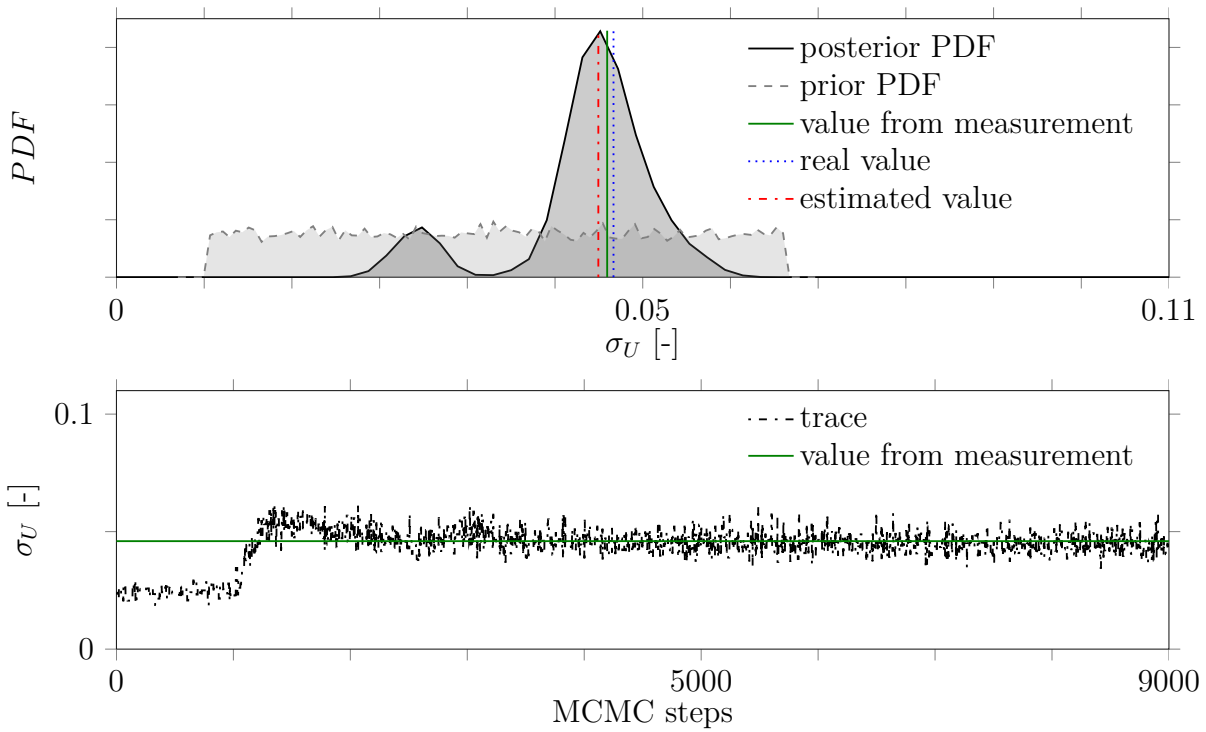

Figure 7: This figure shows the result of the MCMC estimation. The first plot contains the prior and posterior PDF of the hyperparameter σ_U. The second plot visualize the trace.

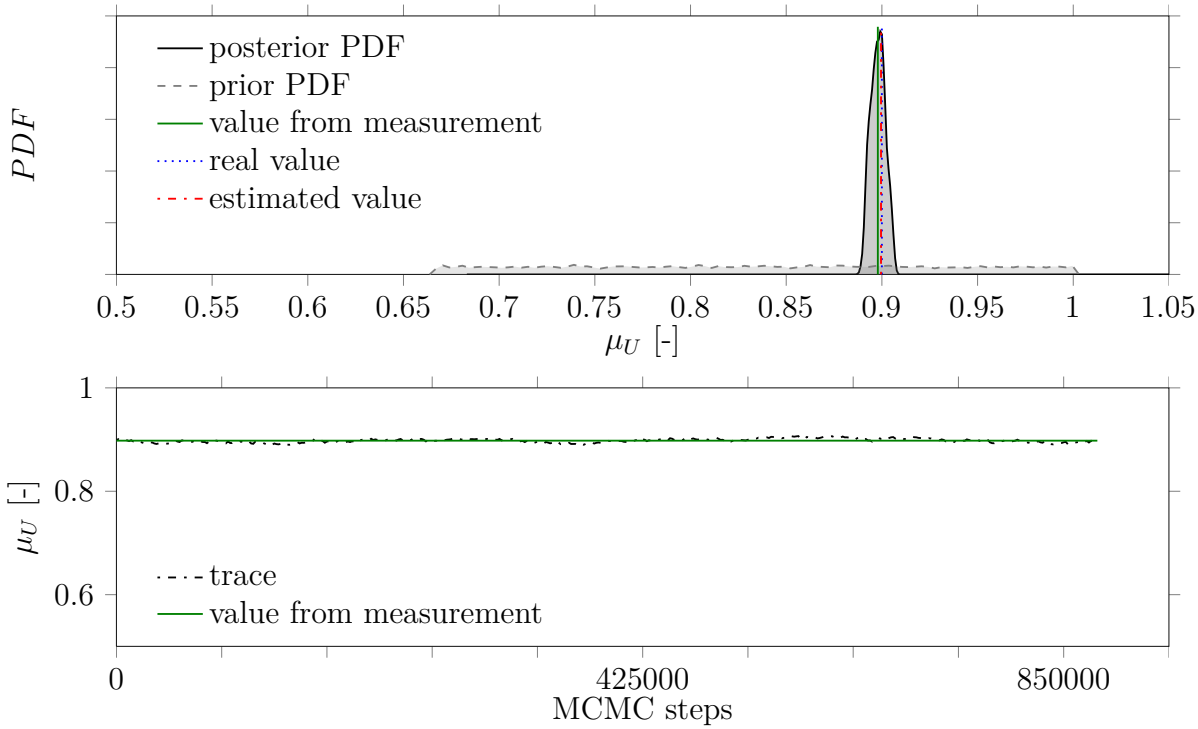

Figure 8: This figure shows the result of the PC surrogate approach. The first plot contains the prior and posterior PDF of the hyperparameter μ_U. The second plot visualize the trace.

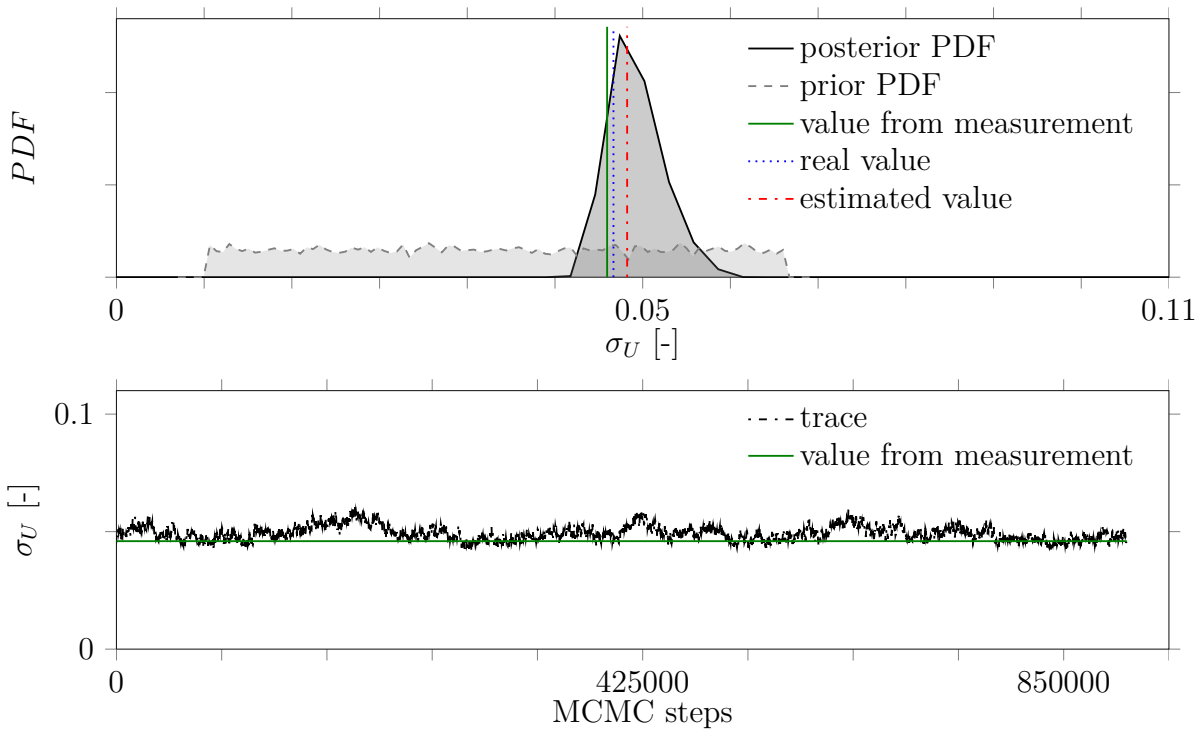

Figure 9: This figure shows the result of the PC surrogate approach. The first plot contains the prior and posterior PDF of the hyperparameter σ_U. The second plot visualize the trace.

Note that the first 10% of the MCMC runs are thrown away to give the Markov Chain time to reach its equilibrium distribution. The results of the posterior distribution using a pure MCMC approach is shown in figure 6 and figure 7. The expected values for the mean and standard derivation of the voltage are well approximated. However, we stopped the iteration at $10,000$ MCMC samples, since the involved computational cost of evaluating the forward model became prohibitive.

The results of the posterior distribution using a MCMC approach based on a PC surrogate model is shown in figure 8 and figure 9. As before the expected values are well approximated and compared to the method without a surrogate model the duration for the estimation was reduced significant. In this case a polynomial order of $p = 3$ was assumed and $Q = 111$ evaluations on the simulation model were performed. Note that increasing the polynomial order in the PC expansion did not provide any further improvement in the approximation of the mean and standard derivation. The remaining uncertainty represents the effect of the discrepancy between simulation model and reality in combination with measurement noise.

5 CONCLUSIONS

We presented a comparison between a PC surrogate model and a MCMC approach for inverse UQ. The used benchmark problem is based on an electric drive test bench hardware. The advantage of this setting is the possibility to define a distribution for an uncertain parameter at the system input. Based on the design of experiment measurement data can be recorded under real-world conditions. This approach gives us the ability to compare the gained results from the method with the assumed, or to put it another way, predefined distributions, which are in general not known.

We focused on the implementation of a MCMC approach, which uses the recorded data with 100 measurement runs. The estimation of the posterior distributions was done using a PC surrogate model in addition to run all evaluations on the simulation model. Due to the fact, that the time for a single simulation run is quite high and therefore the number of evaluations is limited in an industrial context. The duration, which is needed for the estimation of the posterior distributions, can be reduced significantly by using a PC surrogate model. For the pure MCMC run $1,000,000$ evaluations were needed. In contrast, 111 runs on the simulation model were performed to fit the PC surrogate model. Afterwards $100,000,000$ evaluations on the surrogate model for the MCMC approach were done. In addition, the obtained results for both approaches are satisfying.

Our current work is to investigate different types of distributions. Beside that, more uncertain parameters are take into account.

REFERENCES

[1] Barthelmann, V., Novak, E., Ritter, K.: High dimensional polynomial interpolation on sparse grids. Advances in Computational Mathematics **12**(4), 273–288 (2000). DOI 10.1023/A:1018977404843

[2] Cameron, R.H., Martin, W.T.: The Orthogonal Development of Non-Linear Functionals in Series of Fourier-Hermite Functionals. The Annals of Mathematics **48**(2), 385–392 (1947). DOI 10.2307/1969178

[3] Coleman, T.F., Li, Y.: On the Convergence of Reflective Newton Methods for Large-Scale Nonlinear Minimization Subject to Bounds. Mathematical Programming **67**(2), 189–224 (1994)

[4] Coleman, T.F., Li, Y.: An Interior Trust Region Approach for Nonlinear Minimization Subject to Bounds. SIAM Journal on Optimization **6**(2), 418–445 (1996). DOI 10.1137/0806023

[5] Eldred, M., Webster, C., Constantine, P.: Evaluation of nonintrusive approaches for wiener-askey generalized polynomial chaos. 49th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference pp. 1–22 (2008)

[6] Gelfand, A.E., Smith, A.F.M.: Sampling-Based Approaches to Calculating Marginal Densities. Journal of the American Statistical Association **85**(410), 398–409 (1990)

[7] Geman, S., Geman, D.: Stochastic relaxation, gibbs distributions, and the bayesian restoration of images **6**, 721–741 (1984)

[8] Gerstner, T., Griebel, M.: Numerical integration using sparse grids. Numerical Algorithms **18**(3/4), 209–232 (1998). DOI 10.1023/A:1019129717644

[9] Ghanem, R.G., Spanos, P.D.: Stochastic Finite Elements: A Spectral Approach. Springer New York, New York, NY (1991)

[10] Hastings, W.K.: Monte Carlo sampling methods using Markov chains and their applications. Biometrika **57**(1), 97–109 (1970). DOI 10.1093/biomet/57.1.97

[11] Le Maître, O.P., Knio, O.M.: Spectral methods for uncertainty quantification: With applications to computational fluid dynamics. Scientific computation. Springer, Dordrecht and London (2010)

[12] Le Maître, O.P., Knio, O.M., Najm, H.N., Ghanem, R.G.: Uncertainty propagation using Wiener–Haar expansions. Journal of Computational Physics **197**(1), 28–57 (2004). DOI 10.1016/j.jcp.2003.11.033

[13] Metropolis, N., Rosenbluth, A.W., Rosenbluth, M.N., Teller, A.H., Teller, E.: Equation of State Calculations by Fast Computing Machines. The Journal of Chemical Physics **21**(6), 1087 (1953). DOI 10.1063/1.1699114

[14] Novak, E., Ritter, K.: High dimensional integration of smooth functions over cubes. Numerische Mathematik **75**(1), 79–97 (1996). DOI 10.1007/s002110050231

[15] Smolyak, S.: Quadrature and interpolation formulas for tensor products of certain classes of functions. Soviet Mathematics, Doklady **4**, 240–243 (1963)

[16] Tanner, M.A., Wong, W.H.: The Calculation of Posterior Distributions by Data Augmentation. Journal of the American Statistical Association **82**(398), 528–540 (1987). DOI 10.1080/01621459.1987.10478458

[17] The MathWorks Inc.: MATLAB and Optimization Toolbox: Release 2012b, Natick, Massachusetts, United States.

[18] Wiener, N.: The Homogeneous Chaos. American Journal of Mathematics **60**(4), 897–936 (1938). DOI 10.2307/2371268

[19] Xiu, D.: Fast numerical methods for stochastic computations: A review. Communications in Computational Physics **5**(2-4), 242–272 (2009)

[20] Xiu, D., Hesthaven, J.S.: High-Order Collocation Methods for Differential Equations with Random Inputs. SIAM Journal on Scientific Computing **27**(3), 1118–1139 (2005). DOI 10.1137/040615201

[21] Xiu, D., Karniadakis, G.E.: The Wiener-Askey Polynomial Chaos for Stochastic Differential Equations. SIAM Journal on Scientific Computing **24**(2), 619–644 (2002). DOI 10.1137/S1064827501387826