INAUGURAL-DISSERTATION

zur

Erlangung der Doktorwürde

der

Naturwissenschaftlich-Mathematischen Gesamtfakultät

der

RUPRECHT-KARLS-UNIVERSITÄT
HEIDELBERG

vorgelegt von

David Nicholas John, M.Sc.

aus Weingarten

Tag der mündlichen Prüfung:

Uncertainty Quantification for an electric motor inverse problem – tackling the model discrepancy challenge

November 8, 2020

Advisors:

Prof. Dr. Vincent Heuveline

Prof. Dr.-Ing. Carsten Proppe

Chance favors the prepared mind.

– Louis Pasteur (1822-1895) –

Abstract

In the context of complex applications from engineering sciences the solution of identification problems still poses a fundamental challenge. In terms of Uncertainty Quantification (UQ), the identification problem can be stated as a separation task for structural model and parameter uncertainty. This thesis provides new insights and methods to tackle this challenge and demonstrates these developments on an industrial benchmark use case combining simulation and real-world measurement data.

While significant progress has been made in development of methods for model parameter inference, still most of those methods operate under the assumption of a perfect model. For a full, unbiased quantification of uncertainties in inverse problems, it is crucial to consider all uncertainty sources. The present work develops methods for inference of deterministic and aleatoric model parameters from noisy measurement data with explicit consideration of model discrepancy and additional quantification of the associated uncertainties using a Bayesian approach. A further important ingredient is surrogate modeling with Polynomial Chaos Expansion (PCE), enabling sampling from Bayesian posterior distributions with complex simulation models. Based on this, a novel identification strategy for separation of different sources of uncertainty is presented. Discrepancy is approximated by orthogonal functions with iterative determination of optimal model complexity, weakening the problem inherent identifiability problems. The model discrepancy quantification is complemented with studies to statistical approximate numerical approximation error. Additionally, strategies for approximation of aleatoric parameter distributions via hierarchical surrogate-based sampling are developed. The proposed method based on Approximate Bayesian Computation (ABC) with summary statistics estimates the posterior computationally efficient, in particular for large data. Furthermore, the combination with divergence-based subset selection provides a novel methodology for UQ in stochastic inverse problems inferring both, model discrepancy and aleatoric parameter distributions. Detailed analysis in numerical experiments and successful application to the challenging industrial benchmark problem – an electric motor test bench – validates the proposed methods.

Zusammenfassung

Das Lösen von Identifikationsproblemen im Zusammenhang mit komplexen Anwendungen der Ingenieurwissenschaften stellt nach wie vor eine fundamentale Herausforderung dar. Bezüglich der Quantifizierung von Unsicherheiten (UQ) kann das Identifikationsproblem als Trennungsaufgabe zwischen strukturellen Modell- und Parameterunsicherheit formuliert werden. Diese Arbeit bietet neue Erkenntnisse und Methoden zur Bewältigung dieser Herausforderung und demonstriert diese Entwicklungen anhand eines industriellen Anwendungsfalls, welcher Simulation und reale Messdaten kombiniert.

Während in der Entwicklung von Methoden zur Inferenz von Modellparametern erhebliche Fortschritte erzielt wurden, arbeiten doch die meisten dieser Methoden unter der Annahme eines perfekten Modells. Für eine vollständige, unverfälschte Quantifizierung von Unsicherheiten in inversen Problemen ist es entscheidend alle Unsicherheitsquellen zu berücksichtigen. Die vorliegende Arbeit entwickelt Methoden zur Inferenz deterministischer und aleatorischer Modellparameter aus verrauschten Messdaten unter expliziter Berücksichtigung der Modelldiskrepanz und zusätzlicher Quantifizierung der damit verbundenen Unsicherheiten unter Verwendung eines Bayes'schen Ansatzes. Ein weiterer wichtiger Bestandteil ist die Surrogatmodellierung mit polynomieller Chaosentwicklung (PCE), welche das Ziehen von Zufallszahlen aus Bayes'schen Posterior-Verteilungen mit komplexen Simulationsmodellen ermöglicht. Darauf aufbauend wird eine neuartige Identifikationsstrategie zur Trennung verschiedener Unsicherheitsquellen vorgestellt. Die Diskrepanz wird durch orthogonale Funktionen mit iterativer Bestimmung der optimalen Modellkomplexität angenähert, wodurch Schwierigkeiten durch das inhärente Identifikationsproblem geschwächt werden. Die Modelldiskrepanzquantifizierung wird durch Studien zur statistischen Schätzung des numerischen Approximationsfehlers ergänzt. Zusätzlich werden Strategien zur Approximation aleatorischer Parameterverteilungen durch hierarchische Surrogat-basierte Stichprobenverfahren entwickelt. Die vorgeschlagene Methode, basierend auf der Approximativen Bayes'schen Berechnung (ABC) mit zusammenfassenden Statistiken, schätzt die Posterior-Verteilung rechnerisch effizient, insbesondere für große Datenmengen. Darüber hinaus liefert die Kombination mit einer Divergenz-basierten Teilmengenauswahl eine neuartige Methode zur Inferenz von Modelldiskrepanz und aleatorischen Parameterverteilungen für UQ in stochastischen inversen Problemen. Detaillierte Analysen in numerischen Experimenten und erfolgreiche Anwendung auf den herausfordernden, industriellen Anwendungsfall - einen Prüfstand für Elektromotoren - validieren die vorgeschlagenen Methoden.

Acknowledgments

Working on this thesis has been an exciting and eventful time for which I am grateful. I would like to thank the Faculty of Mathematics and Computer Science at Heidelberg University as well as Corporate Research at Robert Bosch GmbH for their infrastructural and professional support.

Above all, I would like to express my sincere gratitude to Prof. Dr. Vincent Heuveline for providing the excellent framework and support for the development of this thesis at the Engineering Mathematics and Computing Lab. Especially, I thank him for consistent guidance and fruitful discussions during the past years, which inspired and sharpened my research. Also many thanks to Prof. Dr.-Ing. Carsten Proppe for his interest in my work and the valuable scientific feedback improving this work. Furthermore, I want to thank Dr. Michael Schick for the generous time for scientific exchange in the years of joint work. His support and friendship helped me in all the time of research and writing. The frequent exchange with my wonderful colleagues within the Virtual Prototypes and Simulation group at Bosch in Renningen and within the Engineering Mathematics and Computing Lab in Heidelberg has been inspiring at all times for which I am very grateful. Especially, many thanks to Philipp Glaser for providing the electric motor test bench, to Dr.-Ing. Daniel Schwarzer for his ongoing support and to Lydia Mehra for her precious kindness. Further, it was a pleasure to work together with Prof. Dr. Claudia Schillings and Livia Stohrer from University of Mannheim. Last but not least, biggest thanks to my family for all the support and love they have shown me during my years of studies. And in particular to my wife Julia and son Louis: Thanks for putting up with me being sat in the office for hours on end. You have been amazing and as promised, I will now start cooking for both of you again.

Contents

Chapter 1.

Introduction

> Life's most important questions are, for the most part, nothing but probability problems.
>
> Pierre-Simon Laplace (1749-1827)

1.1. Challenges in industrial simulation tasks

Scientific computer simulations are a key technology to understand, predict and design natural and engineered systems. This is in particular the case for increasingly complex technical products, where high quality standards and safety critical functions yield high demands on simulation model quality and on accuracy of numerical solution methods. Obvious examples for such complex products are highly autonomous vehicles and robots, but also other electric-mechanical systems like electric motors, batteries, power tools and household devices.

Typically, mathematical models are used to conceptually describe the essential behavior and relationships of complex processes in many applications, such as engineering, physical sciences, biology, medicine, finance and many others. Those models are designed in order to solve a particular task that can not be conducted with the original system, due to time, cost or other limiting constraints. Models however are only approximations of complex real-world phenomena and thus contain sources of uncertainty. This can be due to restrictive assumptions that have to be made, lack of knowledge, uncertain or even unknown parameter values or distributions. Further, conceptual models need to be transferred to algorithmic models or simulation models, to execute experiments on computational devices. This introduces additional uncertainties due to the use of numerical approximations or other simplifications.

In design processes, computer simulations are often used to virtually build a first set of prototypes in order to analyze their validity, efficiency or robustness. Virtual prototyping reduces costs of actually building several iterations of real-world prototypes and optimally allows faster time-to-market. To be successful it is crucial for virtual product development to consider the above mentioned sources of uncertainties in simulation and

also the tolerances, material inhomogeneities and other uncertainties of real products acting in the uncertain open world.

The recent advancements in computational power allow increasingly detailed simulation models and makes their solution ever faster, but uncertainties are still present in most models. These uncertainties pose challenges for the interpretation and reliability of simulation results. By quantifying these uncertainties and taking them into account in mathematical problem descriptions, more refined and reliable predictions can be made. This is what Uncertainty Quantification (UQ) aims to achieve. UQ methods focus on understanding, quantifying and propagating uncertainty in computational simulation of models. A crucial role in this context plays the combination of simulation models with statistical models and data. Often, uncertainties from various sources are categorized into aleatoric and epistemic uncertainties. Aleatoric uncertainties are inherent to the system itself and are stochastic variations that cannot be reduced. Epistemic uncertainties are due to model assumptions, insufficient knowledge, parameterizations or discretizations and can theoretically be reduced, e.g. by including more details or performing more measurements [Smith, 2013, Sullivan, 2015, McClarren, 2018].

The typical framework for UQ is, given a model with inputs and outputs, to identify and quantify sources of uncertainty and then propagate uncertainty from inputs to outputs. Typical engineering questions are then to assess the scattering of the output, the sensitivity of the output scattering w.r.t. the input scattering, probability of failure, optimal design or best-fitting model parameters. Example questions for this risk and reliability assessment might be: How does the output of a technical device vary due to tolerances of certain product parameters? Does the output scattering comply to a specific quality goal or does it lead to events of product failure? What parameters need to be adjusted and can some tolerances even be loosened for cost saving?

A main challenge in UQ is the handling of high computational costs associated with the combination of simulation models and statistical models. A wide range of methods already deal with this issue, e.g. by surrogate modeling [Sullivan, 2015]. Another remedy is the dependence of the UQ analysis on rarely perfect simulation models and on often unavailable, detailed knowledge of model parameters or model parameter distributions, which leads to a discrepancy compared to measurement data.

1.2. Uncertainty in inverse problems

Measurement data can be used to improve the accuracy of simulations by improving the knowledge of model parameters and model parameter distributions. Direct measurement data of a quantity of interest is rarely available. Often measurement data is only indirectly related to the quantity of interest via some model, requiring the solution of an inverse problem. Other classical terms in model and data integration are model calibration, model fitting, model parameter estimation [Sullivan, 2015], system identification

[Söderström and Stoica, 1989] and data assimilation [Reich and Cotter, 2015].

The inverse problem describes the task of finding a model parameter $x \in \mathcal{X}$ for given measurement $y \in \mathcal{Y}$ and model $\mathcal{M} : \mathcal{X} \to \mathcal{Y}$ such that the model output agrees best with the measurement $y = \mathcal{M}(x)$. Generally, this equality does not hold due to

- **parametric uncertainty**, i.e. epistemic uncertainty due to unknown deterministic parameters or aleatoric uncertainty due to (unknown) parameters with intrinsic stochastic variation,
- **observation uncertainty**, i.e. noise and errors that are introduced by observing measurements y,
- **structural model uncertainty**, i.e. lack of knowledge or missing physics in modeling the true system of interest by a model $\mathcal{M}$ and
- **solution method uncertainty**, i.e. errors introduced by implementation and also by numerical approximation schemes.

Hence, one considers $y = \mathcal{M}(x) + \varepsilon$ where ε represents the uncertainty due to measurement noise or model error. Simply inverting $\mathcal{M}$ is not possible as ε is unknown and in general $\mathcal{M}^{-1}$ is non-existent. Typically, inverse problems are ill-posed in the sense of Hadamard. There might exist multiple solutions and stability can be critical, i.e. that small changes in the data imply large changes in the parameters (see e.g. [Dashti and Stuart, 2017, Ch. 4, Example 6]). In order to obtain a well-posed problem regularization is necessary.

This can be done by Bayesian inference which has its origin in the Bayes' theorem[1]. Its idea is to improve knowledge of uncertain parameters by incorporating all available information, namely information about the measurement process and prior knowledge of the uncertain parameters. The Bayesian approach to inverse problems is fundamental in the quantification of uncertainty within applications, involving the blending of mathematical models with data. Further, it allows the incorporation of expert knowledge in the prior. In the absence of suitable data, often expert knowledge is used to construct model parameter distributions for UQ, e.g. by a process called expert knowledge elicitation [O'Hagan, 2019]. The uncertainty of the experts is then represented by a probability distribution, which might be subjective and not reliable enough. The Bayesian approach to inverse problems allows to confirm or update this prior knowledge based on data.

[1] Bayes' theorem is named after Reverend Thomas Bayes (ca. 1701-1761), who formulated a specific case of the theorem during the 1740s in an essay on the [inverse] probability of causes. Later in 1763 it was edited and published in [Bayes, 1763] by Richard Price (1723-1791). However, it was Pierre-Simon Laplace (1749-1827) who rediscovered it independently and gave it its modern mathematical form and scientific application 1774 in [Stigler, 1986, engl. translation] and 1812 in [Laplace, 1812]. The discovery of Bayes theorem, the following controversy in science and achievements assigned to it (e.g. cracking the enigma code) is covered in [McGrayne, 2011], a popular scientific book.

The solution of this probabilistic approach to inverse problems is the so called posterior distribution which describes the updated knowledge of the uncertain parameters, see e.g. [Kaipio and Somersalo, 2005, Stuart, 2010, Dashti and Stuart, 2017]. Since the result of the Bayesian approach to inverse problems is a distribution rather than a single point estimate, one gains additional information about the uncertainty on the estimate in contrast to deterministic approaches.

For complex models, the posterior distribution is usually not available in closed form and needs to be approximated. The approximation can be done by filtering, variational and sampling methods [Stuart, 2010, Gelman et al., 2013]. Markov Chain Monte Carlo (MCMC) methods are often used in practice to compute samples from the posterior distribution [Hastings, 1970, Robert and Casella, 2005, Kaipio and Somersalo, 2005]. An alternative are Approximate Bayesian Computation (ABC) methods which are of advantage whenever the likelihood function is not known in explicit form or when its evaluation is computationally too expensive. The likelihood is then approximated based on the comparison of model simulations with the measurements [Wilkinson, 2013].

Despite the recent developments in advanced sampling methods, the solution of inverse problems with Bayesian inference remains challenging, in particular for high dimensional parameter spaces. If the parameters are aleatoric with unknown probability distributions, then an infinite dimensional problem needs to be addressed. Hierarchical modeling and inference is established, e.g. in [Glaser et al., 2016, Glaser, 2020], but high dimensional and computationally expensive. This requires the development of efficient methods for approximation.

An additional challenge in solving inverse problems is to capture all sources of uncertainty and correctly identify them, often called "identification problem". Here, model discrepancy, in particular, requires special care.

1.3. The special role of simulation model discrepancy

> All models are approximations. Assumptions, whether implied or clearly stated, are never exactly true. **All models are wrong, but some models are useful.** So the question you need to ask is not "Is the model true?" (it never is) but "Is the model good enough for this particular application?"
>
> George E. P. Box (1919 - 2013) in [Box et al., 2009, p.61]

Discrepancy between simulation models and measurement data is almost always present when dealing with modeling and simulation of complex, real-world systems. It occurs over all disciplines and has several synonyms in literature, such as model discrepancy, model error, model-form error, model inadequacy, model uncertainty, structural error and prediction error.

With available measurement data, an arising question, adding on the quote above, is: How can data be used to first asses the model quality and second improve existing models in order to be more useful for the particular application?

In the simulation literature model discrepancy is traditionally addressed during the crucial practice of model verification, validation and calibration. This includes the tasks of characterizing numerical approximation errors (verification), checking if the simulation explains experimental measurements accurately (validation), and if not, refining the model until sufficient accuracy is achieved (calibration) [Oberkampf and Roy, 2010]. UQ is closely related to verification and validation with joint approaches for uncertainty propagation [Roy and Oberkampf, 2011].

In inverse problems, though, quantifying model discrepancy together with other uncertainties is still one of the most fundamental challenges in UQ. While significant progress has been made in development of methods for model parameter calibration, still most of those methods operate under the assumption of a perfect model. For a full quantification of uncertainties in the inverse problem all sources should be considered. Neglecting model discrepancy can lead to biased and overconfident parameter estimates, since model discrepancy is often one of the largest contribution to the overall uncertainty. Considering model discrepancy in inverse problems is crucial in order to obtain realistic calibrations of unknown physical parameters [Brynjarsdóttir and O'Hagan, 2014] and to further obtain a quantification of model quality and reliability, which is important for most engineering tasks.

The Kennedy and O'Hagan framework [Kennedy and O'Hagan, 2001] is one of the first attempts to model and explicitly take account of all the uncertainty sources that arise in the calibration of computer models. In particular, model discrepancy is considered in the Bayesian formulation of the inverse problem with Gaussian Process models. Following work, detailed later in Section 4.1, mainly discusses the core issue with the KO approach: the identification problem, i.e. the problem of distinguishing between effects of the model parameters and the model discrepancy. To improve identifiability, statistical modeling assumptions including sufficient prior information are required [Brynjarsdóttir and O'Hagan, 2014], for instance, via smoothness assumptions and low-degree polynomial expansions for the model discrepancy [Nagel et al., 2020]. However, careful modeling for the model discrepancy prior is required, as its choice has a permanent influence onto the parameter posterior distribution [Tuo and Wu, 2015, Tuo and Wu, 2016].

Overall, considering simulation model discrepancy explicitly in the inverse problem with the additional goal of physically meaningful model parameters is extremely challenging. It requires further development of methods that address the identifiability problem and discrepancy modeling. Further, there is an urge for methods with affordable computational costs. An additional challenge is the consideration of model discrepancy together with aleatoric model parameter inference, which requires the model discrepancy to adapt to the stochastic variations of the parameters.

1.4. Contributions and outline

The driving application in this thesis is a direct current electric motor test bench, introduced in detail in Chapter 2. The use of UQ methods is crucial for the development and production of electric motors in order to meet high quality requirements. Inverse UQ is required to infer unknown, aleatoric model parameters, which are itself required for reliable virtual product development.

Chapter 3 sets up the methodology and theoretical background needed for the methods developed in this thesis. This includes the introduction of Bayesian inference for inverse problems, sampling methods and surrogate modeling. The latter is crucial to make sampling with complex simulation models feasible. A core ingredient for the following methods are Polynomial Chaos Expansion (PCE) approximations for the simulation models.

Chapter 4 introduces methods for simultaneous inference of deterministic model parameters and model discrepancy. The main contributions are:

- A novel probabilistic numerics method for estimation of epistemic uncertainty arising in the solution of nonlinear ODEs. See Section 4.2.
- A novel method for simultaneous estimation of unknown model parameters and model discrepancy from noisy observations for multi output dynamical systems. Adapted from [Nagel et al., 2020], the model discrepancy is hereby modeled by orthogonal functions and extended with an iterative determination of optimal complexity in order to account for identifiability problems. The method is computationally more efficient in handling large number of data points in contrast to Gaussian Processes model discrepancy approaches. See Section 4.3.
- Improvement of sampling with surrogate-based gradients, in Section 4.4.
- Application and detailed analysis of the proposed method for synthetic electric motor data. And successful application to real-world data from the test bench, with challenging determination of optimal complexity. See Section 4.5.

Chapter 5 proposes efficient methods for inference of unknown aleatoric model parameter distributions from noisy observations in complex systems. The parameters are estimated in a hierarchical Bayesian setting. In particular, the main contributions are:

- A novel general methodology, leading to an immense speed-up of ABC sampling by a combination of PCE approximations of the forward problem, sparse grid techniques and the use of summary statistics. The proposed methodology is application-neutral and can be applied to a wide range of problem classes. See Section 5.1.

- Theoretical results in Theorem 3, proving that ABC is stable w.r.t. approximations of the underlying forward model.
- Showcase of the proposed methodology on the electric motor test case with synthetic and real-world data from the test bench. The results of the hierarchical surrogate-based ABC method with summary statistics are compared to an MCMC method using PCE and Laplace based preconditioners for the sampling. In particular, the proposed method demonstrates a performance comparable to MCMC in terms of accuracy while reducing the computational time. The speedup depends on the data size and increases tremendously with larger data.

Chapter 6 then combines aleatoric model parameter and model discrepancy inference with following main contributions:

- A novel methodology called hierarchical surrogate-based discrepancy-corrected inference for aleatoric parameter estimation, by combining the previous methods in a computationally efficient way with divergence-based subset selection and PCE surrogate modeling for the model discrepancy.
- Successful application to the electric motor test bench data and comparison to inference with a model discrepancy surrogate based on reference data.

This work concludes in Section 7. Preliminary results of this thesis were already published in:

- [John et al., 2018a, John et al., 2019b] and in the detailed preprint [John et al., 2018b] for simultaneous inference of one deterministic parameter and model discrepancy;
- [John et al., 2019a] for statistical quantification of epistemic uncertainty arising in the solution of nonlinear ODEs;
- [John et al., 2020] for speed up of aleatoric parameter distribution inference with hierarchical surrogate-based ABC with summary statistics;

Chapter 2.

Application and mathematical modeling

> The electrodynamic action of an induced current opposes equally the mechanical action inducing it.
>
> H. F. Emil Lenz (1804-1865), Lenz's law[1]

The methods considered in this work are applied to a direct current (DC) electric motor test bench. Section 2.1 introduces a basic electric motor simulation model, which is later used for synthetic data generation and detailed analysis. Section 2.2 introduces the electric motor test bench and further detailed modeling. Later on the real world data from the test bench is used to test the considered methods on robustness. Finally, Section 2.3 sets the mathematical description of the problems: first the forward propagation of uncertainty and second – the focus of this work – the inverse problem.

2.1. Basic electric motor model

An electric motor is a rotating electrical machine converting electrical into mechanical energy. Electric motors can either operate on direct (DC) or on alternating current (AC) and are thus different in their power source, construction and control. This work considers DC electric motors, which are powered from a battery or an AC to DC converter. With the magnitude of the applied voltage the motor speed can be varied (for AC motors one needs to adjust the frequency to control the speed). Main elements of DC motors are a stator and a rotor and often the stator is fixed and the rotor is movable, e.g. as in Figure 2.1. The stator creates a magnetic field in which the rotor is located, either by permanent magnets, or by one or more field windings on the stator. The rotor itself is surrounded by the armature winding, consisting of several coils. When electrical current is applied to the armature coil a magnetic field is generated. Due to the permanent magnetic field caused by the stator a physical force, the Lorentz force, affects the rotor. This force causes the rotor to rotate. After a certain rotation, when the magnetic fields are aligned, brushes and a commutator located on the rotor changes the poles, such that the

[1] Lenz's law is a consequence of the conservation of energy. It was first published in [Lenz, 1834] (German), see [Stine, 1923, p.111] for an English translation.

current and therefore the magnetic poles of the rotor reverse their direction. Thus, the rotor remains in its continuous rotation [Toliyat and Kliman, 2004]. For a short history on the invention of electric motors see [Doppelbauer, 2018] and consult the references therein for more details.

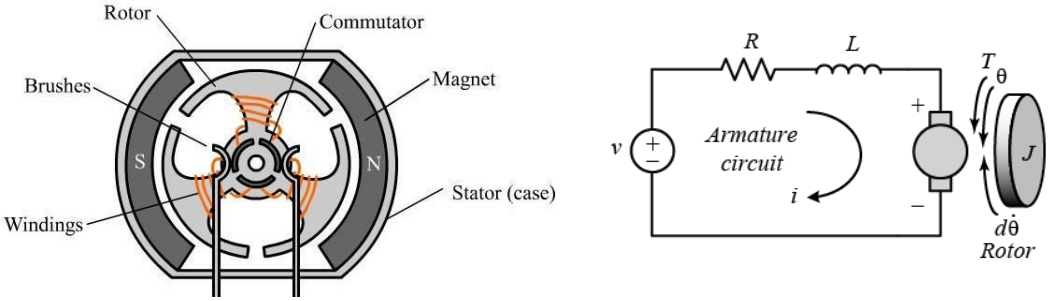


Figure 2.1.: DC electric motor schematic sketch (left) and circuit diagram (right)[2].

Two equations, one representing the electrical and one the mechanical side, are combined for a mathematical description of a DC electric motor system. Variables of interest are current I $[A]$ and voltage V $[V]$ on the electrical side and torque T_m $[Nm]$ and angular velocity, i.e. the rotational speed ω $[rad/s]$ on the mechanical side. The electrical part of a DC motor can be explained by the armature circuit, see Figure 2.1 for a schematic diagram (where θ denotes the angle of rotation and $\dot{\theta} = \omega$). It is determined by

- an applied voltage V that has to be compensated by a resistance R $[\Omega]$, inducing the armature current $I(t)$ and leading to $V = RI(t)$;[3]
- a coil that induces the voltage $V_c = L\dot{I}(t)$, where L $[H]$ is the armature circuit inductance and $\dot{I}(t) = \frac{dI(t)}{dt}$ is the change of current in time;[4]
- and the internal voltage generated by the motor $V_i = c_m\omega$, where c_m $[Vs/rad]$ is a motor constant.

Combining all parts of the armature circuit results in

$$V = RI(t) + L\dot{I}(t) + c_m\omega(t). \tag{2.1}$$

The mechanical part of the DC motor is determined by its torque T_m. It is composed of

- inertia of the motor J $[kg\ m^2]$, which arises only when rotational speed changes, leading to $T_J = J\dot{\omega}(t)$;

[2] Adapted from http://eee-books01.blogspot.com/2015/06/typical-brushed-motor-in-cross-section.html and http://ctms.engin.umich.edu/CTMS/index.php?example=MotorSpeed&section=SimulinkModeling, respectively.
[3] This relationship is called Ohm's law and was proven in 1826 by G.S. Ohm (1789-1854) [Ohm, 1826].
[4] The direction of V_c is given by Lenz's law.

- friction, with friction proportionality constant D $[kg\ m/s]$, which has a direct influence $T_D = D\omega(t)$;
- and by a (constant) load torque T $[Nm]$ which is required to compensate a mechanical load.

Altogether, this leads to

$$T_m = J\dot{\omega}(t) + D\omega(t) + T. \tag{2.2}$$

To combine (2.1) and (2.2), the torque T_m can be expressed by $T_m = c_g I(t)$, where c_g $[Nm/A]$ is another motor constant. For $t > 0$, this leads to the basic electric motor ordinary differential equations (ODEs)

$$\dot{I}(t) = \frac{1}{L}\left(-RI(t) - c_m\omega(t) + V\right), \tag{2.3a}$$

$$\dot{\omega}(t) = \frac{1}{J}\left(c_g I(t) - D\omega(t) - T\right), \tag{2.3b}$$

with initial conditions $I(0) = I_0 \in \mathbb{R},\ \omega(0) = \omega_0 \in \mathbb{R}$ [Toliyat and Kliman, 2004].

This linear ODE system with given constant coefficients can in principle be solved analytically. However, this is generally not the case for most problems. Hence, to be more general we use a numerical method to approximate the solution. Here the ODE system is solved by an explicit Runge-Kutta Method of order 4 (in particular dopri5 (Dormand and Prince), see [Wanner and Hairer, 1991]) with $N_t = 601$ equidistant time steps $t_i, i = 1, \ldots, N_t$, in the time interval $[0, 6]$ seconds. We denote the discretized model, mapping from model parameters $\boldsymbol{x} \in \mathcal{X} \subseteq \mathbb{R}^n$ to discrete approximations of the two states current I and angular velocity ω, by

$$\mathcal{M}_{\triangle t} : \mathcal{X} \to \mathbb{R}^{2\times N_t}. \tag{2.4}$$

In the following we refer to $\mathcal{M}_{\triangle t}$ as simulation model $\mathcal{M}$, by omitting the index $\triangle t$ for notional convenience. Figure 2.2 displays such an approximation of a motor starting

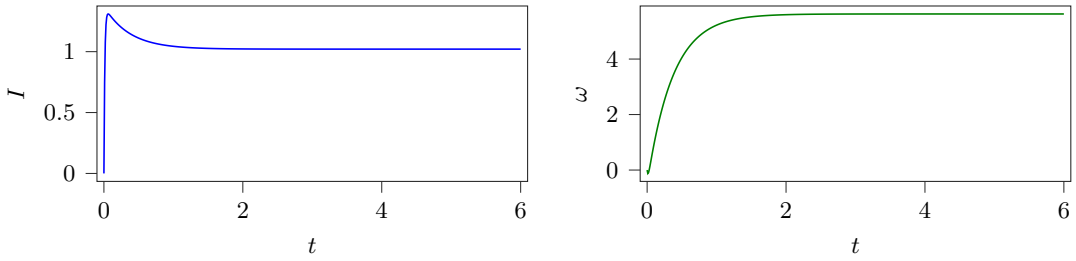


Figure 2.2.: Example trajectory of current I and angular velocity ω for the basic electric motor model (2.3).

from rest, i.e. with $I(0) = 0,\ \omega(0) = 0$, and with a fixed set of parameters $(R = 9, L = 0.11, c_m = 0.5, c_g = 3, D = 0.1, J = 0.1, V = 12, T = 2.5)$.

2.2. Electric motor test bench

This section first describes the set-up of the electric motor test bench, second how measurements are obtained and last the corresponding simulation model. This is partly based on previous work by Glaser et al. [Glaser et al., 2016] and Glaser [Glaser, 2020]. The test bench was also used in own previous work [John et al., 2020], which we follow closely in the upcoming description.

The hardware of the test bench is based on a windshield wiper electric motor with an attached break to mimic mechanical loading. Figure 2.3 presents a schematic diagram. The mounting ⑥ builds the base of the test bench. Starting from left to right a DC power supply ④ is needed to drive the engine. Then there is the electric drive ① composed of the motor and a worm gear. Metal couplings ⑤ connect the shaft components and a torque sensor ② is mounted on the motor shaft. Besides the torque sensor additional sensors are connected with the electric drive to measure thermal characteristics. Furthermore, measurements of rotational speed can be obtained. On the right side of the shaft there is an electromagnetic powder brake ③ that is needed for correctly dealing with a load by minimizing the back-drive ability[5]. The brake opens the possibility to run the motor in different modes. Thereby, different measurements corresponding to the different operating modes of the test bench can be obtained to analyze motor characteristics.

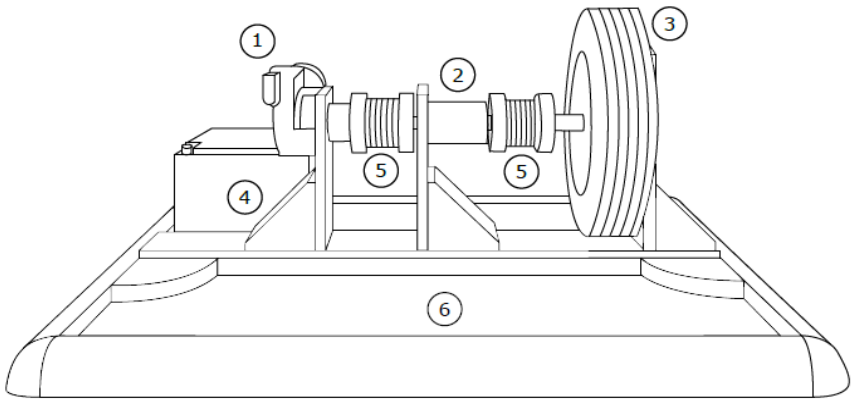


Figure 2.3.: Schematic diagram of the test bench hardware [Glaser et al., 2016].

Due to tolerances, material uncertainties and different suppliers, some components of electric motors have varying properties when coming from the production line. This yields varying characteristics of the motor. The special nature of the test bench is that

[5]Back-drive ability is the degree of ease of which a motor can be driven by its attached load when the power is removed from the motor.

some parameters can be varied on a single set-up in order to mimic some of those variations. This can be done in an automated way and without replacing components of the test bench. Automation is achieved by controlling the test bench with a dSPACE[6] control unit that acts according to a control software on a personal computer. In particular the test bench easily allows to vary the voltage V by a controllable power source and the load torque T by controlling the magnetic powder brake. In reality the load might vary largely, e.g. for a windshield wiper due to effects of wind, rain, snow, ice and dirt. And the voltage might vary due to different batteries, the battery age and usage and due to the energy consumption of other devices. For safety reasons it is important that the electric motor works reliably in all scenarios.

2.2.1. Raw measurement data

To mimic aleatoric parameters we define reference distributions for voltage $\pi(V)$ and torque $\pi(T)$ and draw samples $(V_i, T_i), i = 1, \dots, 200$ from those distributions. Running the test bench sequentially with the sampled values as input results in 200 sets of measurement signals. I.e. for a given sample (V_i, T_i) we start the test bench and record measurement signals of current $\boldsymbol{y}_i^I \in \mathbb{R}^{N_t^{raw}}$ and angular velocity $\boldsymbol{y}_i^\omega \in \mathbb{R}^{N_t^{raw}}$ with sampling frequency 10 kHz for around 10 seconds. The number of data points in each of these raw measurement signal N_t^{raw} is around 10^5. Figure 2.4 displays some of these raw measurement signals.

The raw data requires some preprocessing steps for further proceedings. Before detailing on this, let for notational convenience denote by $(\cdot)$ a placeholder for I or ω interchangeably, e.g. in $\boldsymbol{y}^{(\cdot)}$. Further define following.

Definition 1 (Measurement signal noise estimation)**.** *Let $\boldsymbol{y} = [y(t_1), \dots, y(t_{N_t})] \in \mathbb{R}^{N_t}$ be a noisy measurement signal which is assumed to be stationary in a time interval $[t_a, t_b] \subset [t_1, t_{N_t}]$. Let $J \subset \{1, \dots, N_t\}$ be an index set of time points $t_j \in [t_a, t_b]$ with $j \in J$. Then*

$$\overline{\boldsymbol{y}} = \frac{1}{|J|} \sum_{j \in J} y(t_j). \tag{2.5}$$

denotes the empirical mean of y w.r.t. the time interval $[t_a, t_b]$ and

$$\overline{\sigma} = \sqrt{\frac{1}{|J| - 1} \sum_{j \in J} \left(y(t_j) - \overline{\boldsymbol{y}}\right)^2}, \tag{2.6}$$

denotes the empirical noise standard deviation of y, w.r.t. the time interval $[t_a, t_b]$. (In cases where the measurement signal y is non-stationary a moving average could be used to remove the trend first.)

[6] dSPACE digital signal processing and control engineering GmbH, https://www.dspace.com/

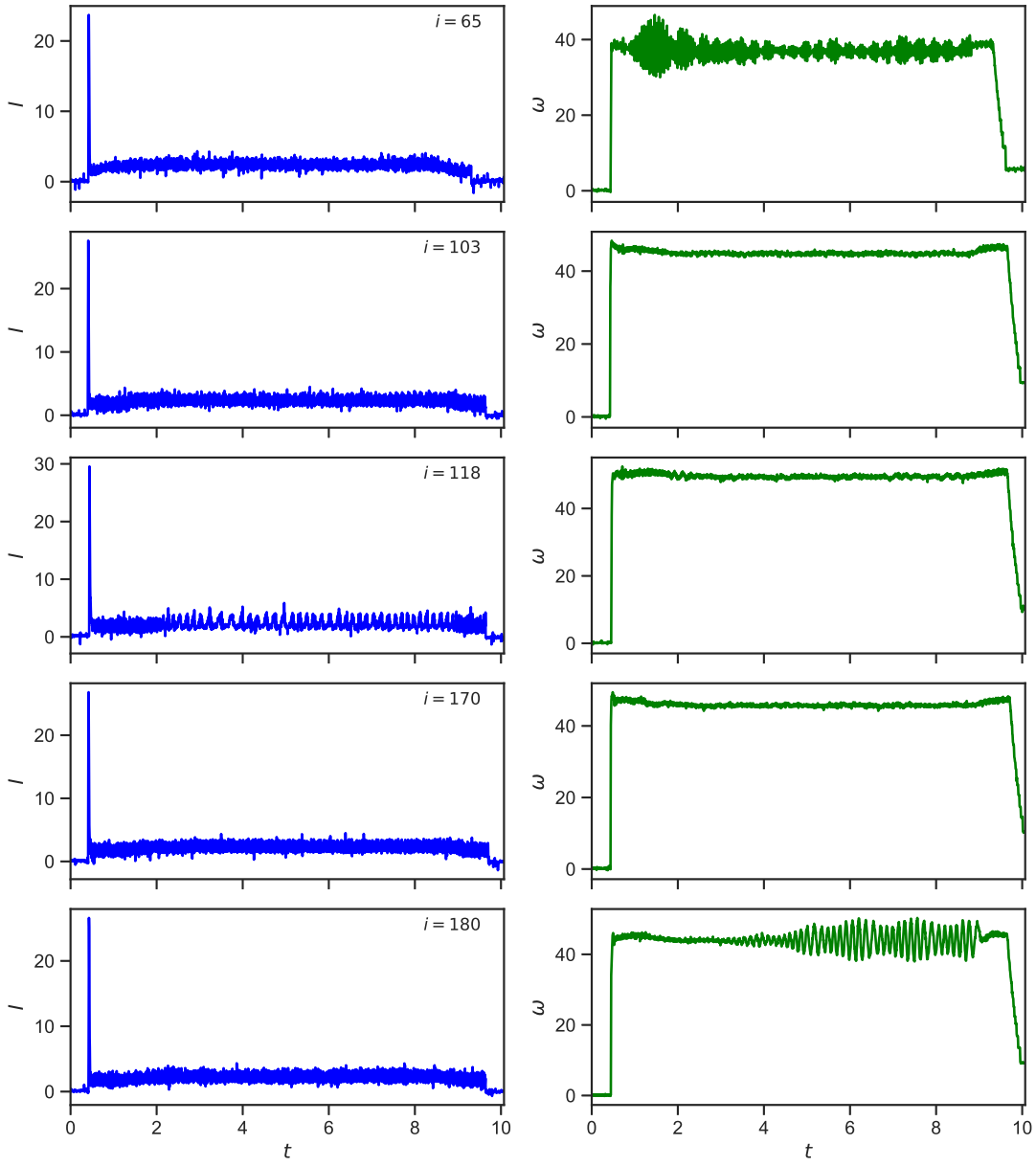


Figure 2.4.: Raw measurement signals of current I and angular velocity ω, for test bench runs with index $i = 65, 103, 118, 170$ and 180 (top to bottom).

Definition 2 (z-scores). *For a set of samples* $\mathcal{S} = \{s_i \in \mathbb{R}, i = 1, \dots, N\}$ *define by*

$$z_i := \frac{s_i - \text{median}(\mathcal{S})}{\text{std}(\mathcal{S})}. \tag{2.7}$$

the z-scores for $i = 1, \dots, N$*, where* $\text{std}(\mathcal{S})$ *is the empirical standard deviation. Optionally, the median can be exchanged by the mean, however the median is less sensitive to outliers.*

The raw data is preprocessed with following steps:

1. **Outlier detection and removal:** In some measurements unwanted oscillations or other patterns occur. Those might be due to the test bench setup and in particular due to the magnetic powder break, which does not reliably apply constant torque for all scenarios. E.g. test bench run with numbers $i = 65$ and $i = 180$ in Figure 2.4 have oscillations in ω and run $i = 118$ has some ripple in I. To find and exclude such runs we compute for each run $i = 1, \dots, 200$ the empirical noise standard deviations $\overline{\sigma_i^{(\cdot)}}$ in the time interval [3,7] seconds (see Definition 1) for the measurement signals of I and ω respectively. Those are displayed in Figure 2.5. Based on $\{\overline{\sigma_i^{(\cdot)}}\}_{i=1,\dots,200}$ we compute the z-scores $z_i^{(\cdot)}$ (see Definition 2) and use thresholds $\tau > 0$ on $z_i^{(\cdot)}$ as a criterion to decide what data should be excluded. This yields the set of excluded indices $\mathcal{I}^{(\cdot)} = \{i : z_i^{(\cdot)} \geq \tau\}$. E.g. for I we exclude all data with $z_i^I \geq 2.5$ and for ω with $z_i^\omega \geq 0.9$, yielding the two sets $\mathcal{I}^I$ and $\mathcal{I}^\omega$, respectively. The union $\mathcal{I} = \mathcal{I}^I \cup \mathcal{I}^\omega$ of the excluded data is marked by red squares in Figure 2.5.

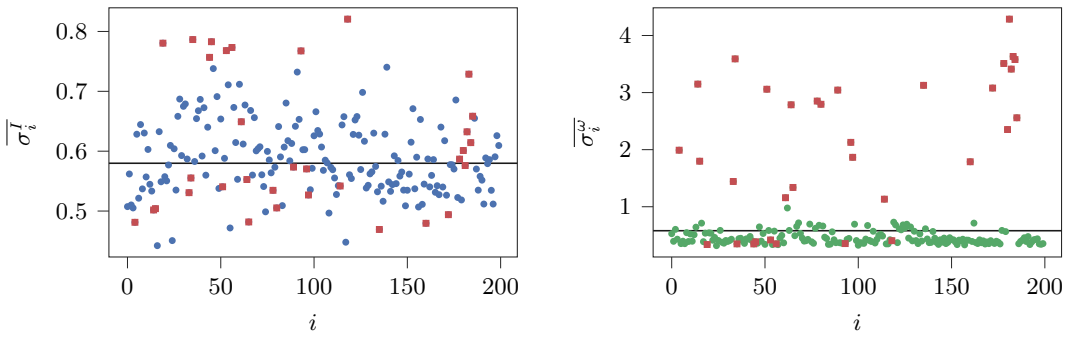


Figure 2.5.: Empirical noise standard deviations $\overline{\sigma_i^{(\cdot)}}$ of the raw test bench measurement data in the stationary time interval [3,7] seconds for runs $i = 1, \dots, 200$. The black line shows the median and the red squares indicate the union $\mathcal{I} = \mathcal{I}^I \cup \mathcal{I}^\omega$ of detected outliers.

2. **Align starting points:** The measurement recording starts roughly 0.4 seconds prior to the ramp up of the applied voltage and corresponding ramp up of the current, see Figure 2.4. Thus we align the starting points of each measurement by discarding the first few observations where only ground noise is happening.

3. **Warm-up phase:** Temperature plays an important role in the behavior of the test bench. Figure 2.6 displays the empirical means $\overline{T_{(\cdot),i}}$, w.r.t. the time interval [0,10] seconds (see Definition 1), for test bench runs $i = 1, \ldots, 200$ of the temperature signals measured at the worm, brush, magnet and winding. The mean of the temperatures increases largely for the first test bench runs and then stabilizes, such that the later runs are all within a similar temperature range. Thus we discard the first few test bench runs in order to account for a warm-up phase of the electric motor and take the last 100 runs, of course with prior removal of the identified outliers.

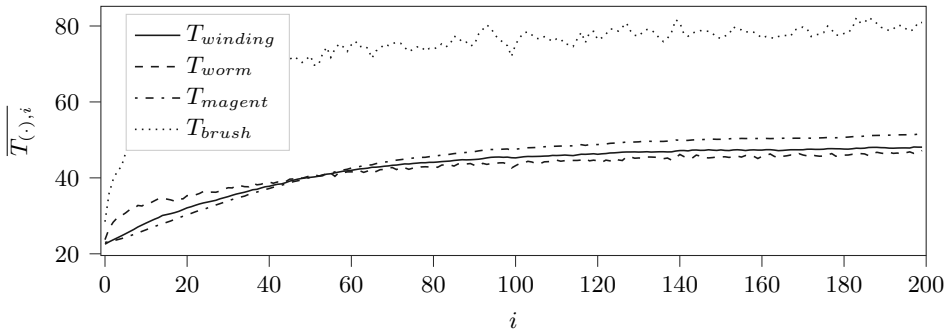


Figure 2.6.: Empirical means $\overline{T_{(\cdot),i}}$, w.r.t. the time interval [0,10] seconds, for test bench runs $i = 1, \ldots, 200$ of the temperatures measured at the worm, brush, magnet and winding.

4. **Filtering:** As detailed in [Glaser, 2020, Ch. 2] the dSPACE control unit is not completely decoupled from the test bench and introduces additional noise components, which have frequencies around 3×10^7Hz and 4×10^5Hz. However the sampling rate of the measurement device is only 10 kHz, thus this additional noise can not be captured fully. To remove the effects a second order Butterworth low pass filter [Butterworth, 1930, Tuzlukov, 2002] is applied on each measurement signal with a cut-off frequency of 5×10^2 Hz. The cut-off frequency is an upper bound on the interesting motor signal characteristics. This filtering step is in particular required when the data is further downsampled, as detailed in the next step.

5. **Downsampling:** To reduce the data size the measurement signals are downsampled from 10kHz to 100 Hz. However this thinning can have major effects on the

resulting data and cautious inspection is required. E.g. downsampling measurement run $i = 170$ (4th row in Figure 2.4) without filtering is very sensitive to the cut-off of the first few data points in Step 2. This is visualized in Figure 2.7 where the mean $\overline{\boldsymbol{y}_{170}^{(\cdot)}}$, w.r.t. the time interval [3,7] seconds, for I and ω is shown versus the number of cut-off time points $c \in \{0, \dots, 20\}$. The effects on $\overline{\boldsymbol{y}_{170}^{(\cdot)}}$ are negligible for ω (less than 0.1% deviation), but drastic and non-acceptable for I, where a deviation of more than 20% occurs. This can be improved by filtering the signals before downsampling.

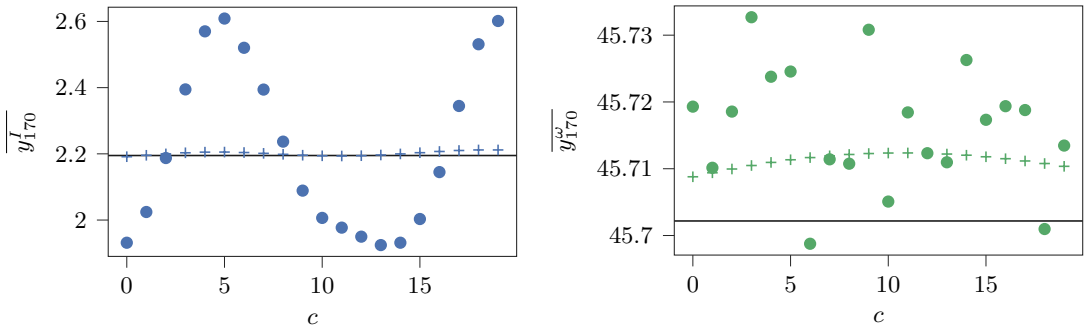


Figure 2.7.: The mean $\overline{\boldsymbol{y}_i^{(\cdot)}}$ for I and ω, w.r.t. the time interval [3,7] seconds, of the downsampled data ($\bullet$) and the filtered and downsampled data ($+$) depends on the number of cut-off time points $c \in \{0, \dots, 20\}$. The mean of the raw data (black line) is (almost sure) independent of c.

6. **Cut-off:** The last step is to cut-off all measurement signals after 6 seconds.

2.2.2. Preprocessed measurement data

After preprocessing the raw data the resulting test bench measurement data set

$$\mathcal{Y} := \{\boldsymbol{y}_i^I, \boldsymbol{y}_i^\omega, i = 1, \dots, N\} \tag{2.8}$$

contains $N = 100$ noisy measurement series of current I and rotational speed ω. For a fixed i the discrete measurement signals

$$\boldsymbol{y}^I := [y^I(t_1), \dots, y^I(t_{N_t})] \in \mathbb{R}^{N_t}, \tag{2.9a}$$
$$\boldsymbol{y}^\omega := [y^\omega(t_1), \dots, y^\omega(t_{N_t})] \in \mathbb{R}^{N_t}, \tag{2.9b}$$

are each of size $N_t = 601$ in the time interval $[0, 6]$ seconds with equidistant time points $(t_1, \dots, t_{N_t}) \in [0, 6]$. Overall, $\mathcal{Y}$ contains $2 \times N \times N_t$ data points. An overview of $\mathcal{Y}$ is given in Figure 2.8.

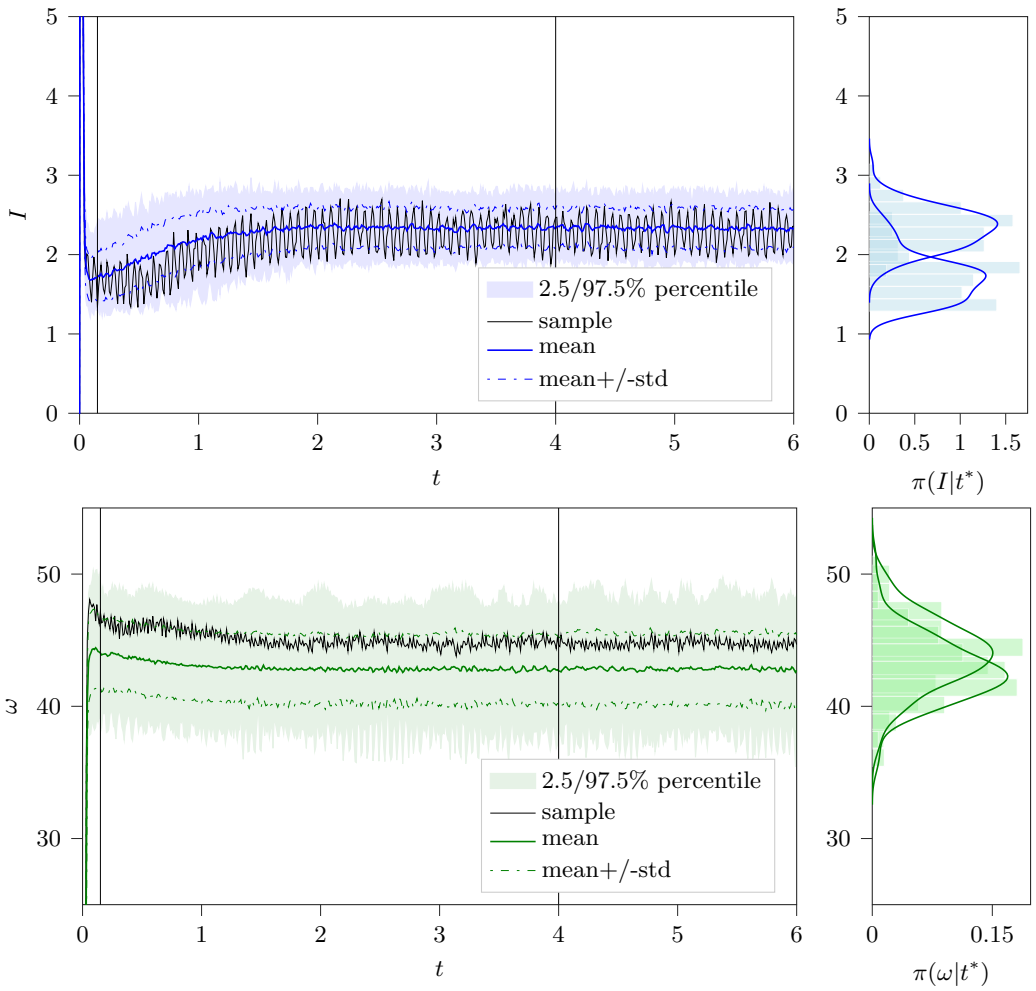


Figure 2.8.: The figures show the preprocessed test bench measurements $\mathcal{Y}$ of the current I and the rotational speed ω for $N = 100$. The area between the 2.5% and 97.5% percentile (shaded), mean+/-standard deviation (dash-dotted) and the mean (solid) of all N measurements are depicted. The black lines show an exemplary noisy sample measurement series. Further, at two time points (vertical lines at $t^* = 0.15$ and $t^* = 4$ seconds) histograms and kernel density estimation plots are displayed on the right hand side. Note, for visualization purpose I values normally reaching $I = 30[A]$ are truncated at $I = 5[A]$.

Reference distributions

So far we have not detailed on the definition of the reference distributions for the aleatoric parameters $X = (V, T)$ in order to generate the data. They are defined as Gaussian with $\pi(V \mid m_V, \sigma_V) = \mathcal{N}(m_V, \sigma_V^2)$ and $\pi(T \mid m_T, \sigma_T) = \mathcal{N}(m_T, \sigma_T^2)$, where the hyper-parameters are set to $m_V = 13.5, m_T = 2.5$, $\sigma_V = 0.7, \sigma_T = 0.2$. With this setting the resulting test bench measurement data set $\mathcal{Y}$ is visualized in Figure 2.8.

To make things clear: In reality the distributions $\pi(V)$ and $\pi(T)$ might not be known. However, this test bench allows to test inference methods on real world data and additionally validate the results on reference distributions and even on reference samples. In this work we constrain ourselves on Gaussian distributions as they are already sufficient to demonstrate the proposed solution methods. Note that the methods introduced in this work can be generalized for other distributions and, of course, can be used for other applications with other parameters, too.

Noise analysis

For a basic analysis of the noise structure in each measurement $\boldsymbol{y}_i^{(\cdot)} \in \mathcal{Y}$ we compute the mean $\overline{\boldsymbol{y}_i^{(\cdot)}}$ (see Definition 1) w.r.t. the stationary time interval $[2, 6]$ seconds and the residuals

$$\boldsymbol{\epsilon}_i^{(\cdot)} = \boldsymbol{y}_i^{(\cdot)} - \overline{\boldsymbol{y}_i^{(\cdot)}} \in \mathbb{R}^{401}, \tag{2.10}$$

respectively for I and ω. With the assumption that the (noise free) underlying signal of I or ω is constant in this time interval, it can be well approximated by $\overline{\boldsymbol{y}_i^{(\cdot)}}$. Consequently, the residuals $\boldsymbol{\epsilon}_i^{(\cdot)}$, interpreted as (approximative) realizations of the underlying observation noise , give insights on the unknown observation noise distribution. Figure 2.9 presents histograms for each $\boldsymbol{\epsilon}_i^{(\cdot)}$ individually and also Gaussian, Laplace and Cauchy distributions fitted (via maximum likelihood estimation) to the union $\{\boldsymbol{\epsilon}_i^{(\cdot)}\}_{i=1,\dots,N}$, respectively for I and ω. Additionally consulting quantile-quantile (Q-Q) plots, suggests the Gaussian distribution to be the best fit (compared to Laplace and Cauchy distributions) for the noise in I and ω, albeit the noise in I has a bi-modal structure. The latter might be due to a phenomena called cogging torque which is due to variations in the magnetic field, see [Islam et al., 2004] and the references therein for further details. An additional cause might be the issues in current recording described in Steps 4 and 5. For simplicity this will be considered as noise in the following and covered within the Gaussian noise assumption. The fitted Gaussian distributions $\mathcal{N}(0, (\sigma^{(\cdot)})^2)$ for $\boldsymbol{\epsilon}_i^{(\cdot)}$ have estimated values $\sigma^I \approx 0.22$ and $\sigma^\omega \approx 0.42$.

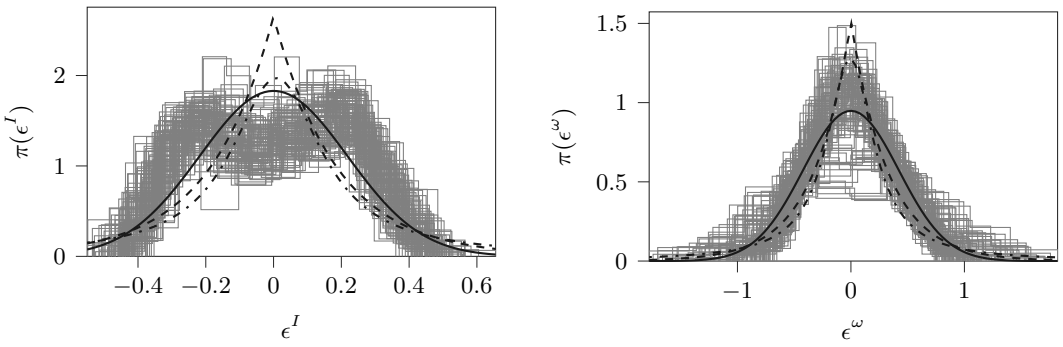


Figure 2.9.: Noise analysis of the preprocessed measurement data $\mathcal{Y}$. The gray histograms present each measurements $\boldsymbol{y}_i^{(\cdot)} \in \mathcal{Y}$ noise $\boldsymbol{\epsilon}_i^{(\cdot)}$ w.r.t. to the stationary time interval $[2, 6]$ seconds, respectively for I and ω. Gaussian (solid black), Laplace (dashed black) and Cauchy (dash-dotted black) distributions are fitted.

2.2.3. Simulation model

In the following two simulation models for the test bench with low and higher complexity are introduced. The basic test bench simulation model differs to the previous basic model (2.3) by modeling the worm gear in the mechanical part of the ODE by

$$\dot{\omega}(t) = \frac{1}{J}\left((c_g I(t) - D\omega(t))\eta - T i_g\right), \tag{2.11}$$

where $\eta \in \mathbb{R}$ is the gear meshing efficiency of the worm gear and $i_g \in \mathbb{R}$ is a constant proportional to the worm gear ratio.

The second test bench model builds up on this and additionally considers a detailed thermal model interconnecting with the electrical and mechanical part. This leads to the introduction of additional parameters describing thermal characteristics and an overall detailed and complex model. In the following we only give a rough overview on the model and refer for details to [Glaser et al., 2016, Glaser, 2020]. The model equations are

$$\dot{I}(t) = \frac{1}{L}\left(-RI(t) - c_m\omega(t) + (V_t(t) - V_{drop})\right), \tag{2.12a}$$

$$\dot{\omega}(t) = \frac{1}{J}\left((c_g I(t) - \tau_{loss} - \tau_{fric})\eta - D\omega(t) - T_t(t) i_g\right), \tag{2.12b}$$

where:

- the resistance R depends linear on the temperature of the coil;
- the motor constant c_m depends nonlinear on temperature of the magnet and on the current $I(t)$; further $c_g \equiv c_m$;

- V_{drop} depends nonlinear on the current $I(t)$;
- the gear meshing efficiency η of the worm gear is a nonlinear function of the lead angle of the worm, the pressure angle and the friction of the worm (depending nonlinearly on the worm temperature);
- τ_{loss} summarizes the hysteresis loss and the eddy current loss, which both depend nonlinear on current $I(t)$ and temperature of the magnet;
- $\tau_{fric} = \tau_{fric,air} + \tau_{fric,motor}$ summarizes the air friction loss $\tau_{fric,air}$ that depends on the angular velocity $\omega(t)$ and the friction loss of the motor $\tau_{fric,motor}$;
- $\tau_{fric,motor}$ is a sum of the friction losses at the bearings, depending on the bearing temperatures and η and the friction loss at the commutator depending on the temperature of the commutator;
- voltage $V_t(t)$ and load $T_t(t)$ for the test bench are considered as time depended, as they are delayed in the hardware until they reach stationary values V and T.

The model parameters are either known from expert knowledge, given by look-up tables or calibrated based on reference measurements with known realizations of V and T. An assumption for the model calibration is that the temperatures of the test bench motor are already on a stationary level. This is in line with the preprocessing Step 3 above, where the first measurements are discarded to account for a warm up phase. Albeit detailed modeling, reality is still not described perfectly by the model and parameter calibration is non-trivial. The latter requires informative data and the solution of a nonlinear optimization problem.

The test bench model is solved by an adaptive numerical integration scheme in the time interval $[0, 6]$. The numerical approximation is then linearly interpolated to $N_t = 601$ equidistant time steps, leading to similar outputs as in the basic model case above. Again, we denote the discretized model, mapping from model parameters $\boldsymbol{x} \in \mathcal{X} \subseteq \mathbb{R}^n$ to discrete approximations of the states I and ω, by

$$\mathcal{M}_{\triangle t} : \mathcal{X} \to \mathbb{R}^{2 \times N_t}. \tag{2.13}$$

As it will be clear from context which model is used, we overload notation and also refer to this numerical approximation as simulation model $\mathcal{M}$, by omitting the index $\triangle t$ for notional convenience.

2.3. Mathematical modeling

This section summarizes and mathematically describes the considered problems. Section 2.3.1 details on forward propagation of uncertainty and Section 2.3.2 on the inverse problem.

2.3.1. Forward propagation of uncertainty

In the previous sections we denoted the numerical approximation of the electric motor ODE systems as simulation model $\mathcal{M}$. To be more specific, for given deterministic model parameters $\boldsymbol{x} \in \mathcal{X} \subseteq \mathbb{R}^n$, the simulation model

$$\mathcal{M} : \mathcal{X} \to \mathbb{R}^{k \times N_t} \tag{2.14}$$

is the operator that numerically approximates the solution of an ODE system of order $k \in \mathbb{N}$ and returns an approximation of the $k = 2$ states (current I and angular velocity ω) at N_t discrete time points $(t_1, \ldots, t_{N_t})$.

In reality some of the simulation model parameters $\boldsymbol{x}$ might be so-called aleatoric parameters, i.e. instead of taking a fixed single value they can fluctuate. Consequently, to reflect these variations in $\boldsymbol{x}$ as well, they are modeled as random variables. Let X_i, $i = 1, ..., n$ be the aleatoric model parameters and $X = (X_1, ..., X_n)$ the vector of all these random variables on an underlying probability space $(\Omega, \mathcal{A}, \mathbb{P})$ with Ω the underlying sample space, $\mathcal{A}$ the sigma algebra and $\mathbb{P} : \mathcal{A} \to [0, 1]$ the probability measure, such that $X : \Omega \to \mathcal{X}$. A realization of X is denoted by $\boldsymbol{x} := X(\omega) = (X_1(\omega), ..., X_n(\omega)) \in \mathcal{X}$ for $\omega \in \Omega$. With the random vector X the simulation model $\mathcal{M}(X)$ becomes random as well with

$$\mathcal{M}(X) : \Omega \to \mathbb{R}^{k \times N_t}. \tag{2.15}$$

A realization of $\mathcal{M}(X)$ is denoted by $\boldsymbol{y} := \mathcal{M}(X)(\omega)$. If $\mathcal{M}$ would be continuous in time, then $\mathcal{M}(X)$ would be a k dimensional stochastic process.

Definition 3 (Forward propagation of uncertainty)**.** *Let $\pi(X)$ be a specific probability distribution of X. The task of forward propagation of uncertainty is now to propagate the uncertainty from X to $\mathcal{M}(X)$ and quantify, i.e. characterize the distribution of $\mathcal{M}(X)$.*

Solutions to this problems are well established, see e.g. [Sullivan, 2015]. Typically Monte Carlo methods are employed for this task by sampling from X and evaluating $\mathcal{M}$ at the samples in order to obtain samples of $\mathcal{M}(X)$. Standard Monte Carlo sampling is very inefficient and if the evaluation of $\mathcal{M}$ is computationally demanding, more efficient methods are required. For example surrogate-based methods, where $\mathcal{M}$ is replaced by a cheaper to evaluate surrogate, see e.g. [Sullivan, 2015]. As there is already an established zoo of methods for forward uncertainty propagation the development of further methods is not the focus of this work, however methods to deal with this problem are helpful to approach the more difficult inverse problem. The solution of the inverse problem often requires numerous evaluations of the forward problem. Consequently, methods used to accelerate the forward problem, for instance, can further be used to solve inverse problems more efficiently.

As already elaborated in the introduction, the forward propagation of uncertainty sensitively depends on the specified probability distribution of X. In reality this is

rarely known in detail and often crude assumptions are made, which of course influences the quality of the quantified distribution of $\mathcal{M}(X)$. Further $\mathcal{M}$ is only an approximation of the reality and thus introduces additionally a simulation-reality mismatch, denoted by simulation model discrepancy.

2.3.2. Inverse problem

Recall the definition and notation of the test bench measurement data $\mathcal{Y}$ in (2.8). First, consider only one test bench run with the measurement series $\boldsymbol{y} := [\boldsymbol{y}^I, \boldsymbol{y}^\omega] \in \mathbb{R}^{2\times N_t}$ and the simulation model $\mathcal{M}$ with simulation model parameters $\boldsymbol{x} \in \mathcal{X}$. Assume that the time points of $\mathcal{M}$ correspond to those of the measurements, which might require additional interpolation within $\mathcal{M}$.

Definition 4 (Inverse problem)**.** *An inverse problem is the task of finding simulation model parameters $\boldsymbol{x}^\dagger \in \mathcal{X}$, such that*

$$\boldsymbol{y} = \mathcal{M}(\boldsymbol{x}^\dagger), \tag{2.16}$$

where the simulation model or forward operator $\mathcal{M}$ is continuous, bounded and sufficiently smooth. The data $\boldsymbol{y}$ is in general limited and noisy.

Section 1.2 already elaborated on the challenges and uncertainties associated with inverse problems. In the following, Chapter 3 details on established solution methods and introduces the Bayesian approach to inverse problems. Then, Chapter 4 present methods to solve the inverse problem by additionally considering simulation model discrepancy.

Now, consider the full data set $\mathcal{Y}$. There, measurement signals have some variations that cannot be explained by measurement noise alone, but are due to unknown aleatoric system parameters X. With modeling of X and the stochastic simulation model $\mathcal{M}(X)$ as in the previous section, we define the following problem.

Definition 5 (Stochastic inverse problem)**.** *A stochastic inverse problem is the problem of finding a probability distribution $\pi(X)$ of the aleatoric simulation model parameters X such that*

$$\Upsilon \overset{d}{=} \mathcal{M}(X), \tag{2.17}$$

where $\Upsilon : \Omega \to \mathbb{R}^{k\times N_t}$ is the true underlying process, which is only known by observations $\mathcal{Y}$. Here $\overset{d}{=}$ denotes the equality in distribution.

This problem, of course, inherits all of the previous challenges. Additionally, despite the number of aleatoric parameters is still finite, it is now an infinite dimensional problem, due to the unknown probability distribution. The parameters to infer are of stochastic nature and not deterministic, thus it is denoted as stochastic inverse problem. Chapter 5 deals with the solution of stochastic inverse problems to infer aleatoric parameter distributions. Building on this, Chapter 6 additionally considers the inference of simulation model discrepancy based on the methods introduced in Chapter 4.

Chapter 3.

Numerical methods for inverse problems

> Man muss immer umkehren. (Invert, always invert.)
>
> Carl Gustav Jacob Jacobi (1804-1851)

This chapter introduces methods for inverse problems that will be used later on. Section 3.1 starts with an overview on Bayesian inference and corresponding posterior approximation methods. A brief review on model selection is given in Section 3.2. Section 3.3 introduces surrogate models such as Polynomial Chaos Expansions and Gaussian processes that are used for acceleration of both forward and inverse problems. Additionally, they are useful for sensitivity analysis, detailed in Section 3.4. In Section 3.5 we conduct a stability analysis for the use of surrogate models in Bayesian inference and finally conclude with related work on inverse problems in Section 3.6.

Note that parts of the text in the following are taken and slightly modified from the authors own work [John et al., 2020].

3.1. Bayesian inference

Recall the inverse problem in Definition 4 where the task is to infer unknown simulation model parameters $\boldsymbol{x}^\dagger \in \mathcal{X}$ from given data $\boldsymbol{y} \in \mathbb{R}^{k \times N_t}$, such that $\boldsymbol{y} = \mathcal{M}(\boldsymbol{x}^\dagger)$. Due to the noise in $\boldsymbol{y}$ and the model error in $\mathcal{M}$ one generally considers

$$\boldsymbol{y} = \mathcal{M}(\boldsymbol{x}^\dagger) + \boldsymbol{\varepsilon}, \tag{3.1}$$

where $\boldsymbol{\varepsilon}$ models the measurement and model error. In an abstract way $\mathcal{M} : \mathcal{X} \to \mathbb{R}^{k \times N_t}$ can be seen as the uncertainty-to-observation map, which consists of the solution operator of the underlying forward model and the observation operator. However, as already mentioned this problem is typically ill-posed in the sense of Hadamard, i.e. existence, uniqueness and stability are not necessarily given. A classical approach to solve the inverse problem is by minimizing the data misfit

$$\min_{\boldsymbol{x} \in \mathcal{X}} \frac{1}{2} \|\boldsymbol{y} - \mathcal{M}(\boldsymbol{x})\|^2. \tag{3.2}$$

In order to obtain a well-posed problem regularization is necessary. One approach is Tikhonov regularization, also known as ridge regression in statistics, where a regularization term $R(\boldsymbol{x})$ is added to the data misfit

$$\min_{\boldsymbol{x}\in\mathcal{X}} \frac{1}{2}\|\boldsymbol{y} - \mathcal{M}(\boldsymbol{x})\|^2 + \lambda R(\boldsymbol{x}), \tag{3.3}$$

for a $\lambda > 0$ [Kaipio and Somersalo, 2005]. An example is $R(\boldsymbol{x}) = \|\hat{\boldsymbol{x}} - \boldsymbol{x}\|^2$ for some norm $\|\cdot\|$ and a fixed $\hat{\boldsymbol{x}} \in \mathcal{X}$.

The Bayesian approach to inverse problems yields a natural regularization of ill-posed problems due to the modeling and definition of a probability distribution for the measurement and model error $\boldsymbol{\varepsilon}$ and a prior distribution for the unknown parameters $\boldsymbol{x}$. With this the Bayesian regularization has a clear interpretation in terms of statistics of $\boldsymbol{\varepsilon}$ and $\boldsymbol{x}$. In contrast to this, as some say, the regularization in classical optimization is somewhat arbitrary [Stuart, 2010]. Furthermore and more importantly, the Bayesian approach is popular for uncertainty quantification in inverse problems, as it allows to quantify the underlying uncertainties, such as in the unknown parameters $\boldsymbol{x}$. It delivers an unique probability measure containing information about the relative probability of different parameters $\boldsymbol{x}$ given the data $\boldsymbol{y}$, whereas classical optimization often delivers just one local minimizer. Note that classical optimization also allows some kind of uncertainty estimation for local minimizers with Cramér-Rao bounds based on the Fisher information, for instance, but this is with the assumption of Gaussianity for the estimator. We refer the reader to [Kaipio and Somersalo, 2005, Stuart, 2010, Dashti and Stuart, 2017] for more details on Bayesian inverse problems.

We model the unknown parameters $\boldsymbol{x} \in \mathbb{R}^n$ as random variables, characterized according to a given prior distribution μ_0 and assume that the measurement and model error $\boldsymbol{\varepsilon}$ is independent of $\boldsymbol{x}$ and (for simplicity) normally distributed, i.e. $\boldsymbol{\varepsilon} \sim \mu_e = \mathcal{N}(0, \Gamma)$ with $\Gamma \in \mathbb{R}^{kN_t \times kN_t}$ symmetric, positive definite. The solution of the Bayesian inverse problem is then the posterior distribution, the conditional distribution of the unknown parameters $\boldsymbol{x}$ given the observations $\mathcal{Y}$. The posterior distribution can be characterized via Bayes' formula.

Theorem 1. *[Stuart, 2010] Assume that the least-squares potential $\Phi : \mathcal{X} \times \mathbb{R}^{k\times N_t} \to \mathbb{R}$ with $\Phi(\boldsymbol{x}; \mathcal{Y}) = \frac{1}{2}\|\mathcal{Y} - \mathcal{M}(\boldsymbol{x})\|_\Gamma^2$ is measurable w.r.t. the product measure $\nu_0(\mathrm{d}\boldsymbol{x}, \mathrm{d}\mathcal{Y}) = \mu_0(\mathrm{d}\boldsymbol{x})\mu_e(\mathrm{d}\mathcal{Y})$ and that*

$$Z = \int_{\mathbb{R}^n} \exp(-\Phi(\boldsymbol{x}; \mathcal{Y}))\mu_0(\mathrm{d}\boldsymbol{x}) > 0 \tag{3.4}$$

for $\mathcal{Y}$ μ_e-a.s.. Then, the conditional distribution $\mu^{\mathcal{Y}}$ of $\boldsymbol{x}|\mathcal{Y}$ exists, is absolutely continuous w.r.t. the prior distribution μ_0 and the Radon-Nikodym derivative is given by

$$\mu^{\mathcal{Y}}(\mathrm{d}\boldsymbol{x}) = \frac{1}{Z}\exp(-\Phi(\boldsymbol{x}; \mathcal{Y}))\mu_0(\mathrm{d}\boldsymbol{x}) \tag{3.5}$$

for $\mathcal{Y}$ μ_e*-a.s..*

In the finite dimensional setting, Bayes' theorem can be formulated via the corresponding Lebesgue densities as

$$\pi(\boldsymbol{x} \mid \mathcal{Y}) = \frac{1}{Z}\pi_e\left(\mathcal{Y} - \mathcal{M}(\boldsymbol{x})\right)\pi_0(\boldsymbol{x})\,, \tag{3.6}$$

where π_0 and π_e denote the Lebesgue densities of the prior and measurement distributions, respectively. In the following we correspond to $\pi(\boldsymbol{x} \mid \mathcal{Y})$ as posterior distribution of $\boldsymbol{x}$ given $\mathcal{Y}$. The measurement distribution $\pi_e\left(\mathcal{Y} - \mathcal{M}(\boldsymbol{x})\right)$ is also denoted as likelihood and often the notation $\pi_e\left(\mathcal{Y} \mid x\right)$ is used interchangeably. The likelihood combines the physical model $\mathcal{M}$ with the statistical model of the measurement and model error μ_e. $Z = \int_{\mathbb{R}^n} \pi_e\left(\mathcal{Y} - \mathcal{M}(\boldsymbol{x})\right)\pi_0(\boldsymbol{x})\,d\boldsymbol{x}$ is called marginal likelihood or evidence and works as a normalizing constant. It requires the solution of a high dimensional, complex integral which is computational challenging. Thus it is usually omitted and only following is considered

$$\pi(\boldsymbol{x} \mid \mathcal{Y}) \propto \pi_e\left(\mathcal{Y} - \mathcal{M}(\boldsymbol{x})\right)\pi_0(\boldsymbol{x}). \tag{3.7}$$

Note that in the Bayesian sense, the prior and posterior distributions are rather to be interpreted as degree-of-belief or plausibility of deterministic parameters $\boldsymbol{x}$, than as actual probability distributions. They reflect the epistemic uncertainty. I.e. the prior distribution reflects the belief of plausible values for $\boldsymbol{x}$ before data is observed, e.g. based on previous knowledge or an experts opinion. And the posterior distribution is then an update of this prior belief based on the available data $\boldsymbol{y}$ and the specified likelihood.

Generally, the posterior distribution $\pi(\boldsymbol{x}|\boldsymbol{y})$ is intractable and one can not sample from it directly. In particular due to the solution operator of the underlying forward map involved in the characterization of the posterior distribution, we usually rely on approximation methods.

3.1.1. Maximum a posteriori estimator and Laplace approximation

For some real world applications, the approximation of the whole posterior distribution is computationally challenging. Point estimators or approximations via simpler distributions such as Gaussian distributions are very common in practice to reduce the overall computational effort. One commonly used point estimator is the maximum likelihood estimator (MLE)

$$\boldsymbol{x}_{\mathrm{MLE}} = \arg\max_{\boldsymbol{x}\in\mathcal{X}} \pi_e\left(\mathcal{Y} - \mathcal{M}(\boldsymbol{x})\right), \tag{3.8}$$

which, however, does not take into account the prior. In contrast to the MLE, the maximum a posteriori (MAP) estimator

$$\boldsymbol{x}_{\mathrm{MAP}} = \arg\max_{\boldsymbol{x}\in\mathcal{X}} \pi(\boldsymbol{x} \mid \mathcal{Y}) = \arg\max_{\boldsymbol{x}\in\mathcal{X}} \pi_e\left(\mathcal{Y} - \mathcal{M}(\boldsymbol{x})\right)\pi_0(\boldsymbol{x}) \tag{3.9}$$

is defined as the point $\boldsymbol{x}$ with highest posterior density, consequently taking the prior into account. Note that there is the possibility that the MAP estimate is not unique or does not exist. For definition of the MAP in the infinite dimensional setting, we refer to [Dashti et al., 2013, Helin and Burger, 2015].

The Bayesian approach can be linked to classical optimization methods. For example the least squares solution of the data misfit (3.2) corresponds to the MLE. And the least squares solution of the regularized data misfit (3.2) corresponds to the MAP, if prior distribution and regularization term correspond. See [Stuart, 2010] and also [Dashti and Stuart, 2017, Ch. 4.3] for further reading. In the latter reference the link between the MAP and a Tikhonov-Phillips regularized least squares problem is shown for Gaussian priors.

In case of large data or informative data, the posterior distribution often shows a concentrated behavior in a small region of the parameter domain. In this setting, the posterior can be well represented by the Laplace approximation $\tilde{\mu}^{\mathcal{Y}}_{\text{Laplace}}$

$$\mu^{\mathcal{Y}} \approx \tilde{\mu}^{\mathcal{Y}}_{\text{Laplace}} := \mathcal{N}(\boldsymbol{x}_{\text{MAP}}, C), \tag{3.10}$$

a Gaussian distribution with mean $\boldsymbol{x}_{\text{MAP}}$ and covariance $C = H^{-1}(\boldsymbol{x}_{\text{MAP}})$, where $H(\boldsymbol{x}_{\text{MAP}})$ denotes the Hessian of the log-posterior density at the MAP. We refer to [Schillings et al., 2019] for more details.

Using a Quasi-Newton method to optimization Equation 3.9 yields the MAP and additionally an approximation to the inverse Hessian at the MAP. By using the Symmetric-Rank-1 (SR1) Hessian update strategy theoretic results for convergence to the true Hessian are available, see e.g. [Schillings and Schwab, 2016] and the references therein.

3.1.2. Markov Chain Monte Carlo (MCMC)

Markov Chain Monte Carlo (MCMC) methods approximate the posterior distribution $\pi(\boldsymbol{x} \mid \mathcal{Y})$ by generating a Markov chain with the posterior as limit distribution (under suitable conditions). MCMC methods are often seen as the gold standard for Bayesian inverse problems. Most common are the Metropolis Hastings algorithm, the Gibbs sampler, the Hybrid Monte Carlo algorithm (HMC) and their respective variants. We briefly introduce the Metropolis Hastings MCMC (MH-MCMC) algorithm and refer the reader to [Robert and Casella, 2005, Gelman et al., 2013, Dashti and Stuart, 2017] and the references therein for further details on MCMC methods. For details on the gradient-based No-U-turn sampler (NUTS) see [Hoffman and Gelman, 2014].

For a proposal distribution (or candidate generating kernel) $q : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}_+$ MH-MCMC generates samples that approximate $\pi(\boldsymbol{x}) := \pi(\boldsymbol{x} \mid \mathcal{Y})$ according to Algorithm 1. Usually we cannot evaluate the posterior distribution directly, thus it is replaced in the acceptance ratio α by the right hand side of Equation (3.6), where luckily the evidence

Algorithm 1: Metropolis Hastings MCMC algorithm [Hastings, 1970]

1. Pick the initial value $\boldsymbol{x}_1 \in \mathbb{R}^n$ and set $j = 1$.
2. Draw $\boldsymbol{x}^\star$ from proposal distribution $q(\boldsymbol{x} \mid \boldsymbol{x}_j)$ and calculate the acceptance ratio
$$\alpha(\boldsymbol{x}_j, \boldsymbol{x}^\star) = \min\left(1, \frac{\pi(\boldsymbol{x}^\star) q(\boldsymbol{x}_j \mid \boldsymbol{x}^\star)}{\pi(\boldsymbol{x}_j) q(\boldsymbol{x}^\star \mid \boldsymbol{x}_j}\right).$$
3. Draw $u \in [0, 1]$ from an uniform probability density.
4. If $\alpha(\boldsymbol{x}_j, \boldsymbol{x}^\star) \geq u$, accept $\boldsymbol{x}^\star$ and set $\boldsymbol{x}_{j+1} = \boldsymbol{x}^\star$, else reject and set $\boldsymbol{x}_{j+1} = \boldsymbol{x}_j$.
5. If $j = J$, the desired sample size, stop, else increment j and return to 2.

cancels in the fraction, such that

$$\alpha(\boldsymbol{x}_j, \boldsymbol{x}^\star) = \min\left(1, \frac{\pi_e\left(\mathcal{Y} \mid \boldsymbol{x}^\star\right) \pi_0(\boldsymbol{x}^\star) q(\boldsymbol{x}_j \mid \boldsymbol{x}^\star)}{\pi_e\left(\mathcal{Y} \mid \boldsymbol{x}_j\right) \pi_0(\boldsymbol{x}_j) q(\boldsymbol{x}^\star \mid \boldsymbol{x}_j}\right). \tag{3.11}$$

A standard option for q is the Gaussian random walk proposal distribution

$$q(\boldsymbol{x} \mid \boldsymbol{x}_j) \propto \exp\left(-\frac{1}{2\gamma^2}\|\boldsymbol{x} - \boldsymbol{x}_j\|^2\right), \tag{3.12}$$

with the step size $\gamma > 0$. It is symmetric such that its evaluation cancels in $\alpha(\boldsymbol{x}_j, \boldsymbol{x}^\star)$. The value of γ determines how fast the algorithm explores the distribution and how many proposals are rejected. For small γ the exploration is very slow and the number of accepted samples is usually high. For larger values of γ the exploration is faster, but the number of accepted samples usually decreases. Additionally, if the initialization $\boldsymbol{x}_1 \in \mathbb{R}^n$ in Step 1 of Algorithm 1 is in a very low probability region or even outside of the posterior distribution, then it takes a while (also depending on γ) until MH-MCMC actually generates samples from the posterior. The beginning of the chain is often called burn-in and should be removed as it poorly represents the distribution, see [Kaipio and Somersalo, 2005, Ch. 3.6.2] for an illustration. Further, one should be aware of that due to construction the samples are correlated, which might require a thinning of the samples for further proceeding. This comes on top of the already slow convergence rate $O(1/\sqrt{J})$ of standard Monte Carlo. All these points influence the efficiency and convergence of MH-MCMC and require a careful selection of the step size γ and the stopping criterion, i.e. the desired sample size J.

This work uses an implementation of MH-MCMC with Gaussian random walk proposal distribution based on the Python package PyMC3 [Salvatier et al., 2016], where the step size γ is automatically adjusted in an initial tuning phase.

The Laplace approximation $\tilde{\mu}^{\mathcal{Y}}_{\text{Laplace}}$ from Equation 3.10 can be further used to accelerate sampling by increasing sampler efficiency, see e.g. [Schillings and Schwab, 2016, Schillings et al., 2019, Rudolf and Sprungk, 2018]. The estimation of the MAP $\boldsymbol{x}_{\text{MAP}}$

can be used as starting point $\boldsymbol{x}_1$ for the sampler, which clearly reduces the burn-in phase of the algorithm. And in case of MH-MCMC with Gaussian random walk proposal, the inverse Hessian of the log-posterior density at the MAP C can be used as initialization of the proposal distribution covariance. I.e. replace the proposal distribution $\mathcal{N}(\boldsymbol{x}_j, \gamma^2\mathbf{I})$ by $\mathcal{N}\left(\boldsymbol{x}_j, \gamma^2 C)\right)$, where $\boldsymbol{x}_j$ denotes the current state of all unknown variables that shall be inferred, $\mathbf{I} \in \mathbb{R}^{n\times n}$ the identity matrix and γ the step size parameter.

In general MH-MCMC with Gaussian random walk proposal is not dimension-independent. We refer to [Sprungk, 2018] and [Hu et al., 2017] for further details on methods to increase the performance of the algorithm in high dimensional spaces, such as the preconditioned Crank–Nicolson MCMC algorithm.

3.1.3. Approximate Bayesian Computation (ABC)

Approximate Bayesian Computation (ABC) methods are another alternative to approximate the posterior distribution $\pi(\boldsymbol{x} \mid \mathcal{Y})$, which are of advantage whenever the likelihood function is not known in explicit form or if its evaluation is computationally too expensive. The likelihood is then approximated based on the comparison of model simulations with the measurements [Wilkinson, 2013]. Note that the following introduction to ABC methods is based on the authors own work [John et al., 2020].

ABC methods have been extensively used in population genetics and became known by one of the first publications in 1997 by Tavaré et al. [Tavaré et al., 1997]. Besides the rejection algorithms, ABC methods have been further extended to "likelihood-free" MCMC by Marjoram et al. [Marjoram et al., 2003], sequential Monte Carlo by Sisson et al. [Sisson et al., 2007], probabilistic approximate rejection ABC by Wilkinson [Wilkinson, 2013] and Gibbs Sampling by Wilkinson et al. [Wilkinson et al., 2010]. In ABC methods it is common practice to summarize the measurements by so-called summary statistics in order to improve the efficiency, see e.g. [Fearnhead and Prangle, 2012, Najm and Chowdhary, 2016]. However, the identification of proper summary statistics containing sufficient information for inference is a difficult task, see e.g. [Cam, 1964] and [Nunes and Balding, 2010, Fearnhead and Prangle, 2012, Barnes et al., 2012, Prangle et al., 2014]. For a comparative review of dimension reduction methods in ABC we refer to [Blum et al., 2013]. For a detailed introduction to ABC methods we refer to the book [Sisson et al., 2018]. A direct comparison between MCMC and ABC based on summary statistics is performed by Beaumont et al. [Beaumont et al., 2002] in the context of population genetics.

An advantage of ABC algorithms is that they are intuitive and easy to implement. Algorithm 2 belongs to the classical family of ABC algorithms based on rejection sampling. Let $d(\cdot,\cdot) : \mathbb{R}^{k\times N_t} \times \mathbb{R}^{k\times N_t} \to \mathbb{R}$ be a distance measure between simulated values and observed data. The tolerance $\delta \in \mathbb{R}_+$ determines the accuracy of the algorithm. The accepted prior samples are independent and identically distributed samples of the approximated posterior density $\hat{\pi}\left(\boldsymbol{x} \mid \mathcal{Y}, d(\mathcal{Y}^{sim}, \mathcal{Y}) \leq \delta\right)$.

Algorithm 2: Approximate Rejection [Wilkinson, 2013]

1. Sample $\boldsymbol{x}$ from $\pi_0(\cdot)$.
2. Sample $\boldsymbol{\varepsilon}$ from $\pi_e(\cdot)$ and set $\mathcal{Y}^{sim} := \mathcal{M}(\boldsymbol{x}) + \boldsymbol{\varepsilon}$.
3. Accept $\boldsymbol{x}$ if $d(\mathcal{Y}^{sim}, \mathcal{Y}) \leq \delta$; return to 1.

Two key challenges of ABC methods are finding an appropriate distance measure d as well as a suitable value for the tolerance δ. For $\delta \to \infty$, every draw from the prior is accepted and Algorithm 2 only reproduces the prior. On the other hand, for $\delta = 0$, simulations of the model are only accepted if they are identical to the observed data, resulting in samples of the exact posterior density $\pi(\boldsymbol{x} \mid \mathcal{Y})$. Generally, the accuracy increases with decreasing tolerance and consequently the samples of the approximated posterior get closer to the true posterior distribution. However, the overall acceptance rate decreases with decreasing δ and as a consequence computational effort is higher. Thus, the tolerance has to be chosen as trade-off between computational capacity and accuracy. In practice, a strictly positive tolerance $\delta > 0$ is necessary, since the probability that $\mathcal{Y}^{sim} = \mathcal{Y}$ is in most non-trivial models very low or even impossible due to model miss-specification or measurement noise [Sunnaker et al., 2013].

One possibility to increase the acceptance rate and thus the efficiency is to extend Algorithm 2 by using summary statistics $S(\cdot)$. This summary can be a set of statistics, i.e. $S(\cdot) = (S_1(\cdot), ..., S_d(\cdot))$, $d \in \mathbb{N}$, where the summaries $S_i(\cdot)$ for $i = 1, ..., d$ are for instance the mean, standard deviation or any other statistical measure of the data. Ideally, the summary statistics are sufficient for the parameter $\boldsymbol{x}$ and capture all of the information such that

$$\pi(\boldsymbol{x} \mid S(\mathcal{Y})) \overset{d}{=} \pi(\boldsymbol{x} \mid \mathcal{Y}). \tag{3.13}$$

An insufficient summary would be a further approximation of the original problem. Therefore, the selection of statistical functions has to be done carefully, such that $S(\mathcal{Y})$ is a sufficient representation of the original data $\mathcal{Y}$. Algorithm 3 generates samples that

Algorithm 3: Approximate Rejection with summary statistics [Barber et al., 2015]

1. Sample $\boldsymbol{x}$ from $\pi_0(\cdot)$.
2. Sample $\boldsymbol{\varepsilon}$ from $\pi_e(\cdot)$, set $\mathcal{Y}^{sim} := \mathcal{M}(\boldsymbol{x}) + \boldsymbol{\varepsilon}$ and summarize $S(\mathcal{Y}^{sim})$.
3. Accept $\boldsymbol{x}$ if $d\left(S(\mathcal{Y}^{sim}), S(\mathcal{Y})\right) \leq \delta$; return to 1.

approximate $\pi(\boldsymbol{x} \mid S(\mathcal{Y}))$. The acceptance ratio is now proportional to the probability that the summary statistics of a simulation exactly fits the summary statistics of the observed data and this is usually more probable than matching the whole data exactly. In total, summary statistics reduce the dimension of the data and therefore increase the probability of accepting parameters which makes the algorithm computationally more

efficient. Summary statistics are used in many practical applications, although it is hard to tell if the summaries are sufficient. See e.g. [Najm and Chowdhary, 2016] for more details.

3.1.4. Sequential Monte Carlo ABC (SMC ABC)

Sequential Monte Carlo (SMC) methods define a sequence of intermediate distributions, starting with the prior distribution and iteratively transforming to the posterior distribution [Del Moral et al., 2006]. SMC can be understood as an extension of importance sampling, where, instead of re-weighting the accepted samples to account for differences between prior and posterior distribution, several populations are sampled by using the weights known from importance sampling [Sisson et al., 2007, Lintusaari et al., 2016].

Algorithm 4 is a version of such a SMC ABC method, which is also known as Population Monte Carlo (PMC) method. It is in principle an iterative application of Algorithm 3, where in each population p (resp. iteration) the acceptance rate increases, allowing decreasing tolerance thresholds $\delta^{(p)}$. In Step 5.4. ϕ is most often a standardized

Algorithm 4: Sequential Monte Carlo ABC [Beaumont et al., 2009, Marin et al., 2012]

Set iteration number $p = 1$.

1. Sample $\boldsymbol{x}_i^{(1)}$ from $\pi_0(\cdot)$ for $i = 1, ..., M$.
2. Sample $\boldsymbol{\varepsilon}_i$ from $\pi_e(\cdot)$, set $\mathcal{Y}_i^{sim} := \mathcal{M}(\boldsymbol{x}_i^{(1)}) + \boldsymbol{\varepsilon}_i$ and summarize $S(\mathcal{Y}_i^{sim})$ for $i = 1, ..., M$.
3. Accept $\boldsymbol{x}_i^{(1)}$ if $d\left(S(\mathcal{Y}_i^{sim}), S(\mathcal{Y})\right) \leq \delta^{(1)}$ and set $w_i^{(1)} = 1/m$, where $m \leq M$ is the number of accepted samples.
4. Calculate $\Sigma^{(1)} = 2 \cdot \mathrm{Var}\left(\{\boldsymbol{x}_i^{(1)} | i = 1, ..., m\}\right)$.

For iteration number $p = 2, ..., P$ do

5.1. Sample $\boldsymbol{x}_i^*$ from $\{\boldsymbol{x}_j^{(p-1)} | j = 1, ..., m\}$ with probabilities $w_j^{(p-1)}$ for $i = 1, ..., M$.

5.2. Sample $\boldsymbol{x}_i^{(p)}$ from $\mathcal{N}(\boldsymbol{x}_i^*, \Sigma^{(p-1)})$ for $i = 1, ..., M$.

5.3. Sample $\boldsymbol{\varepsilon}_i$ from $\pi_e(\cdot)$, set $\mathcal{Y}_i^{sim} := \mathcal{M}(\boldsymbol{x}_i^{(p)}) + \boldsymbol{\varepsilon}_i$ and summarize $S(\mathcal{Y}_i^{sim})$ for $i = 1, ..., M$.

5.4. Accept $\boldsymbol{x}_i^{(p)}$ if $d\left(S(\mathcal{Y}_i^{sim}), S(\mathcal{Y})\right) \leq \delta^{(p)}$ and set

$$w_i^{(p)} \propto \pi_0\left(\boldsymbol{x}_i^{(p)}\right) \left(\sum_{j=1}^{m} w_j^{(p-1)} \phi\left(\frac{\boldsymbol{x}_i^{(p)} - \boldsymbol{x}_j^{(p-1)}}{\sqrt{\Sigma^{(p-1)}}} \right) \right)^{-1}.$$

5.5. Calculate $\Sigma^{(p)} = 2 \cdot \mathrm{Var}\left(\{\boldsymbol{x}_i^{(p)} | i = 1, ..., m\}\right)$.

Gaussian or a Student's t density [Beaumont et al., 2009]. For population $p = 1$ a set of samples $\boldsymbol{x}_i^{(1)}$ are drawn from an initial prior distribution $\pi_0(\cdot)$ which are accepted if the summary statistics of the simulated and the observed data are smaller than threshold $\delta^{(1)}$ with respect to the distance measure d. In the following iterations $p = 2, ..., P$ the previous samples $\boldsymbol{x}_i^{(p-1)}$ are used to define a mixed density of normal distributions weighted by $w_j^{(p-1)}$ known from importance sampling. This mixed density is used to generate the next set of samples, $\boldsymbol{x}_i^{(p)}$, which again are only accepted if the summary statistics of the simulated and the observed data are smaller than $\delta^{(p)}$ w.r.t. d. With each p the threshold $\delta^{(p)}$ decreases and consequently the mixed density approaches the true posterior distribution allowing to generate approximated samples from it directly.

3.2. Model selection

Model selection is required in the case where several physical model instances $\mathcal{M}_i\,, i \in \mathbb{N}$, but also if several variants of the statistical error model μ_e and the prior μ_0 are available. The task at hand is to find among all competing models the best model with respect to the available data $\boldsymbol{y}$. We briefly introduce methods that analyze the residual , compare the predictive capability and select models via Bayes' rule. For further reading we refer to [Claeskens and Hjort, 2008].

Residual analysis

Residual analysis is a basic but useful class of techniques to evaluate the goodness of a fitted regression model. Basically the consistency of the underlying error assumptions are rechecked. Let $\boldsymbol{x}^\dagger$ be the solution of (3.1) and denote by $\hat{\boldsymbol{y}} = \mathcal{M}(\boldsymbol{x}^\dagger)$ the prediction. The residuals

$$\boldsymbol{r} = \boldsymbol{y} - \hat{\boldsymbol{y}} \tag{3.14}$$

are then interpreted as estimates of the model error $\boldsymbol{\varepsilon}$ and are used to validate the assumptions concerning $\boldsymbol{\varepsilon}$. This is usually done via graphical plots and statistical tests, requiring careful examination in order to make a judgment on the model-data fit. For $\boldsymbol{\varepsilon}$ modeled as i.i.d. Gaussian with zero mean, one needs to examine if $\boldsymbol{r}$ is a valid sample of $\boldsymbol{\varepsilon}$ by checking the empirical mean, independence, constant variance and Gaussianity. For further details we refer to [Draper and Smith, 1998]

Predictive capability

To compare the predictive capabilities of models, tools like the Akaike (AIC), Deviance (DIC), and Watanabe-Akaike Information Criteria (WAIC) as well as Leave-One-Out cross-validation (LOO) are well suited and widely used. They all estimate pointwise

out-of-sample prediction accuracy from a fitted Bayesian model. Gelman et al. [Gelman et al., 2014] review the AIC, DIC and WAIC and put them into a Bayesian predictive context to better understand their practical application. They state that predictive accuracy can be used as one of the criteria to evaluate, understand and compare models, underpinned with examples. For cross-validation, the data is partitioned into training and holdout sets repeatedly, iteratively fitting the model with the training set and evaluating the fit with the holdout data. As the name implies, the holdout sets for LOO contain just one data point. WAIC (also known as Widely-Applicable Information criterion) and LOO are asymptotically equal [Watanabe, 2010] and utilize the log-likelihood evaluated at the posterior samples of the model parameters to estimate the pointwise out-of-sample prediction fit. They are advantageous over simpler approaches such as AIC and DIC, but involve additional computational steps. Vehtari et al. [Vehtari et al., 2017] introduce fast and stable computations for LOO and WAIC and also an efficient computation of LOO from MCMC samples, using Pareto-smoothed importance sampling (PSIS) for correction.

Bayesian model selection

The Bayesian inference scheme can be applied at the model class level in order to select the best model out of a (finite) set $\boldsymbol{\mathcal{M}} = \{\mathcal{M}_i\,, i \in \mathbb{N}\}$ of candidate model classes, where the statistical model for $\boldsymbol{\varepsilon}$ might vary as well with i. For this we formulate (3.6) to explicitly denote the condition on the model $\mathcal{M}_i$ and its parameters $\boldsymbol{x}_i \in \mathbb{R}^{n_i}$ by

$$\pi(\boldsymbol{x}_i \mid \mathcal{Y}, \mathcal{M}_i) = \frac{1}{Z_i}\pi_e\left(\mathcal{Y} \mid \boldsymbol{x}_i, \mathcal{M}_i\right)\pi_0(\boldsymbol{x}_i \mid \mathcal{M}_i)\,, \tag{3.15}$$

where the evidence is

$$Z_i = \pi\left(\mathcal{Y} \mid \mathcal{M}_i\right) = \int_{\mathbb{R}^{n_i}} \pi_e\left(\mathcal{Y} \mid \boldsymbol{x}_i, \mathcal{M}_i\right)\pi_0(\boldsymbol{x}_i \mid \mathcal{M}_i)\, d\boldsymbol{x}_i. \tag{3.16}$$

Bayes' theorem on model class level reads

$$\pi(\mathcal{M}_i \mid \mathcal{Y}, \boldsymbol{\mathcal{M}}) = \frac{1}{\pi\left(\mathcal{Y} \mid \boldsymbol{\mathcal{M}}\right)}\pi\left(\mathcal{Y} \mid \mathcal{M}_i\right)\pi(\mathcal{M}_i \mid \boldsymbol{\mathcal{M}})\,, \tag{3.17}$$

where $\pi\left(\mathcal{Y} \mid \mathcal{M}_i\right)$ denotes the model class likelihood, evidence or Bayes factor and is equal to (3.16). If the prior distribution for each model class $\pi(\mathcal{M}_i \mid \boldsymbol{\mathcal{M}})$ is uniform, as it usually the case, it is sufficient to compute each models evidence $Z_i = \pi\left(\mathcal{Y} \mid \mathcal{M}_i\right)$. The highest Z_i identifies the model that best explains the data. The values Z_i can also be used for Bayesian model averaging by linearly combining $\mathcal{M}_i$ weighted with Z_i.

An advantage of Bayesian model selection according to [Beck and Muto, 2007, Simoen et al., 2013] is that it automatically enforces model parsimony (also well known as Occam's razor), i.e. less complex models are preferred. Unfortunately, approximation of

the evidence is non-trivial in most cases, as it requires the computation of a high dimensional and complex integral. Most posterior approximation methods, such as MCMC or ABC, neglect the evidence as it is just a normalizing constant. However, some consider methods only fully Bayesian, if the evidence is computed as well.

Estimation of the evidence with standard Monte Carlo by sampling from the prior is in general very inefficient, as only a small fraction of prior samples lie in high likelihood region. Other methods use asymptotic approximations, MCMC samples of the posterior, or multi-level MCMC methods (e.g. Transitional MCMC [Ching and Chen, 2007]), see [Simoen et al., 2013, Ch. 3.4] and the references therein. Sequential Monte Carlo (SMC) methods approximate the posterior by a series of annealed sequences [Kirkpatrick et al., 1983, Neal, 2001, Beck and Au, 2002] $\pi(\boldsymbol{x} \mid \mathcal{Y})_\beta \propto \pi_e(\mathcal{Y} \mid \boldsymbol{x})^\beta \pi_0(\boldsymbol{x})$, for a temperature $\beta \in [0, 1]$, from the prior $\beta = 0$ to the posterior $\beta = 1$, where the evidence is estimated as a by-product, see [Del Moral et al., 2006] and the references therein. Another prominent approach is nested sampling [Skilling, 2006]. It relies on sampling within a sequence of hard constraint on the likelihood value (in contrast to the soft annealing in e.g. SMC), leading to "nested" contours of the likelihood. The evidence is immediately obtained and samples from the posterior distribution are an optional by-product.

The evidence takes into account the prior and depends, unfortunately, sensitively on the specified prior for each model. For informative data, changes in the prior have only minor influence on the posterior distribution, but can have major influence on the value of the evidence. Thus, based on the approximation quality of the estimated evidence values, evidence needs to be handled carefully for selecting a model.

3.3. Surrogate models

If the computational model is expensive to evaluate, reliable and faster to evaluate surrogate models are a popular option in order to speed up computation for the task at hand. In particular sampling methods require numerous evaluations of the simulation model and consequently, surrogate models that replace the original model are often essential to make sampling feasible.

In principle a model $\mathcal{M}$ is approximated by a surrogate $\hat{\mathcal{M}}$ which is cheaper to evaluate. For construction of $\hat{\mathcal{M}}$ some evaluations of $\mathcal{M}$ are required. The points where $\mathcal{M}$ is evaluated are often constructed by a Design of Experiment (DoE). A DoE either randomly draws points from the domain of the model parameters (e.g. Monte Carlo, Quasi-Monte Carlo and Latin Hypercube sample points) or follows a deterministic rule to select points (e.g. some kind of a grid) or iteratively learns points w.r.t. maximizing some information gain (active learning).

In principle every universal approximator is suitable for this task, see Table 3.1 for a non-exhaustive list. For UQ tasks two methods have proven to be notably useful in the context of UQ and are thus widely employed. Namely Polynomial Chaos Expansions

(PCEs) and Gaussian processes (GPs), which are introduced in the following.

Method	Parameters
Artificial Neural Networks	Number of hidden layers, number of neurons per layer, activation function, learning rate, etc.
Gaussian processes	Type of kernel, trend, nugget
Polynomial Chaos Expansions	Basis of polynomials, order of polynomials, sparsity
Support Vector Machines	Regularization, loss function, type of kernel, unregularized terms

Table 3.1.: A non non-exhaustive list of universal approximators and their respective parameters. See e.g. [Bishop, 2006, Sullivan, 2015] for details.

3.3.1. Polynomial Chaos Expansions (PCE)

Polynomial Chaos Expansions (PCE) approximate random variables by expanding them as a series of orthogonal polynomials. Together with efficient methods to compute a finite set of the expansion coefficients, it can be used to reliably approximate considerably smooth simulation models $\mathcal{M}$ with a surrogate that has almost no evaluation costs. The use of PCEs emerged in the late 1980s for stochastic finite element methods [Ghanem and Spanos, 2003], where the problems are discretized both in physical space (by finite element methods) and in random space (by PCE). This requires so-called *intrusive* solution methods that are able to solve the coupled physical-random system. *Non intrusive* methods in contrast only require repeated evaluations of the existing simulation model $\mathcal{M}$ and are thus suited for black-box models[1]. In this work we focus on the non intrusive approach, such that a broad class of problems independent of the underlying equations can be addressed. This is in particular important in industry to ease re-usability and reduce costs.

In 1938, Wiener [Wiener, 1938] introduced the so-called "Homogeneous Chaos"[2] as the span of Hermite polynomial functionals of a Gaussian process. "Polynomial Chaos" (PC) is defined as a member of this set. It is a Fourier-Hermite series expansion, for which Cameron and Martin proved in 1947 [Cameron and Martin, 1947], that the orthogonal representation converges to any square-integrable functional. In context of stochastic processes, this implies that the Homogeneous Chaos converges to any stochastic process of second order, which was put in a broader framework involving more general probability distributions, referred to as "generalized Polynomial Chaos" (gPC) by Xiu and

[1] The categorization in intrusive and non intrusive is permeable from a mathematical point of view and rather needs to be understood with respect to implementation, see [Giraldi et al., 2015] for details.

[2] The term chaos in this context is related to the randomness of a Brownian motion or white noise and does not relate to the state of a dynamical system.

Karniadakis in 2002 [Xiu and Karniadakis, 2002].

Consider, without loss of generality, a real-valued random variable $Y = Y(\omega)$ according to some probability space $(\Omega, \mathcal{A}, \mathbb{P})$. Furthermore, assume that $Y \in L^2(\Omega)$. For a sequence of centered, normalized and mutually orthogonal Gaussian random variables $\{\xi_i\}_{i=1}^{\infty}$ define $\hat{\Gamma}_p$ to be the space of polynomials in $\{\xi_i\}_{i=1}^{\infty}$ with polynomial degree less than or equal to $p \in \mathbb{N}$. Furthermore, define $\Gamma_p \subset \hat{\Gamma}_p$ to be the set of polynomials, which belong to $\hat{\Gamma}_p$ and which are orthogonal to $\hat{\Gamma}_{p-1}$. The space spanned by Γ_p shall be denoted by $\tilde{\Gamma}_p$. Then, the Cameron and Martin theorem [Cameron and Martin, 1947] yields:

$$\hat{\Gamma}_p = \hat{\Gamma}_{p-1} \oplus \tilde{\Gamma}_p, \quad L^2(\Omega) = \bigoplus_{i=0}^{\infty} \tilde{\Gamma}_i. \tag{3.18}$$

The subspace $\tilde{\Gamma}_p$ of $L^2(\Omega)$ is called the $p-$th Homogeneous Chaos, and Γ_p the Polynomial Chaos of order p. A square-integrable random variable Y can then be represented via:

$$Y = a_0 H_0 + \sum_{i_1=1}^{\infty} a_{i_1} H_1(\xi_{i_1}) + \sum_{i_1}^{\infty} \sum_{i_2=1}^{i_1} a_{i_1 i_2} H_2(\xi_{i_1}, \xi_{i_2}) + \dots . \tag{3.19}$$

$H_n(\xi_{i_1}, \dots, \xi_{i_n})$ denotes the multi-dimensional Hermite polynomial of order n in terms of the multi-dimensional independent standard Gaussian random variable $\xi := (\xi_{i_1}, \dots, \xi_{i_n})$. To simplify notation, the multi-indices $i_1, \dots, i_n$ can be enumerated by a single index yielding:

$$Y = \sum_{i=0}^{\infty} b_i \psi_i(\xi), \tag{3.20}$$

using a one-to-one mapping between the PCE coefficients a and b and the basis functionals H and ψ. Instead of using Hermite polynomials, one can also use other orthogonal polynomials depending on the distribution of ξ. This is then called generalized Polynomial Chaos expansion, see [Xiu and Karniadakis, 2002]. It is originally known as Wiener-Askey polynomial chaos, due to the use of orthogonal polynomials from the Askey-scheme, where the orthogonality is with respect to the inner product related to the probability distribution of ξ. Table 3.2 displays some orthogonal polynomials and their relation to probability distributions. Numerical experiments with these combinations demonstrate (exponential) convergence according to the Cameron-Martin Theorem [Cameron and Martin, 1947], whereas the use of non-optimal type of polynomials slows down the convergence rate [Xiu and Karniadakis, 2002].

Remark. Another popular expansion, the Karhunen-Loève decomposition, can be interpreted as a particularly simple form of PCE [O'Hagan, 2013, Remark 17]. Notably, it is a special case of PCE with only linear dependence on the input random variables, whereas PCE, in general, allows polynomial dependence.

	Probability distribution	Polynomial	Support
Continuous	Gaussian	Hermite	$(-\infty, \infty)$
	Gamma	Laguerre	$[0, \infty)$
	Beta	Jacobi	$[a, b]$
	Uniform	Legendre	$[a, b]$
Discrete	Poisson	Charlier	$\{0, 1, 2, \dots\}$
	Binomial	Krawtchouk	$\{0, 1, 2, \dots, N\}$
	Negative Binomial	Meixner	$\{0, 1, 2, \dots\}$
	Hypergeometric	Hahn	$\{0, 1, 2, \dots, N\}$

Table 3.2.: The relation of probability distributions and orthogonal polynomials of the Askey-scheme according to [Xiu and Karniadakis, 2002].

3.3.2. Approximation of simulation models with PCE

Now consider the randomized simulation model $\mathcal{M}(X) : \Omega \to \mathbb{R}^{k \times N_t}$ with $X : \Omega \to \mathbb{R}^n$. Assume that X is independent such that the probability density function factors $\pi(\boldsymbol{x}) = \prod_{i=1}^{n} \pi_i(x_i)$. For the sake of simplicity let by $f = \mathcal{M}_{(i,j)} : \mathcal{X} \to \mathbb{R}$ for $i \in \{1, 2\}$ and $j \in \{1, \dots, N_t\}$ denote only one scalar output of the model $\mathcal{M}$, as the following holds component wise. Now, we expand $f(X)$ as PCE

$$f(X) = \sum_{\boldsymbol{\alpha} \in \mathbb{N}^n} b_{\boldsymbol{\alpha}} \psi_{\boldsymbol{\alpha}}(X), \tag{3.21}$$

by adopting the multi index notation with $\boldsymbol{\alpha} = [\alpha_1, \dots, \alpha_n] \in \mathbb{N}^n$. The polynomials $\psi_{\boldsymbol{\alpha}}(X) = \prod_{i=1}^{n} \psi_{\alpha_i}(X_i)$ are products of univariate orthogonal polynomials, where α_i determines the polynomial degree of ψ_{α_i}. The $\psi_{\boldsymbol{\alpha}}$ are orthogonal with respect to the probability density function of X, i.e.

$$\langle \psi_{\boldsymbol{\alpha}}(X), \psi_{\boldsymbol{\beta}}(X) \rangle = \mathbb{E}[\psi_{\boldsymbol{\alpha}}(X) \psi_{\boldsymbol{\beta}}(X)] = \int_{\Omega} \psi_{\boldsymbol{\alpha}}(X(\omega)) \psi_{\boldsymbol{\beta}}(X(\omega)) d\mathbb{P}(\omega) \tag{3.22}$$

$$= \int_{\mathcal{X}} \psi_{\boldsymbol{\alpha}}(\boldsymbol{x}) \psi_{\boldsymbol{\beta}}(\boldsymbol{x}) \pi(\boldsymbol{x}) d\boldsymbol{x} = \langle \psi_{\boldsymbol{\alpha}}(X), \psi_{\boldsymbol{\alpha}}(X) \rangle \delta_{\boldsymbol{\alpha}\boldsymbol{\beta}} \tag{3.23}$$

which holds in particular dimension wise, due to construction. The random polynomials $\{\psi_{\boldsymbol{\alpha}}(X)\}_{\boldsymbol{\alpha} \in \mathbb{N}^n}$ form an orthogonal basis of $L^2(\Omega)$ and the deterministic polynomials $\{\psi_{\boldsymbol{\alpha}}(X(\omega))\}_{\boldsymbol{\alpha} \in \mathbb{N}^n}$ form an orthogonal basis of $L^2(\mathcal{X})$. Consequently, the realization of the PCE expansion of $f(X)$ can be seen as a deterministic expansion of $f(\boldsymbol{x})$ with deterministic $\boldsymbol{x} \in \mathbb{R}^n$. For computational feasibility, the PCE must be truncated

$$f(X) \approx \sum_{\boldsymbol{\alpha} \in A(p)} b_{\boldsymbol{\alpha}} \psi_{\boldsymbol{\alpha}}(X) = \boldsymbol{b}^T \boldsymbol{\psi}(X), \tag{3.24}$$

where typically a total order truncation set $A(p) := \{\boldsymbol{\alpha} \in \mathbb{N}^n : 1 \leq |\boldsymbol{\alpha}| \leq p\}$ for a finite $p \in \mathbb{N}$ is used. p is then the maximal polynomial degree in one dimension. This truncation leaves $P+1 = (p+n)!/(p!n!)$ terms in the expansion. For other truncation schemes, e.g. the hyperbolic truncation, see [Blatman and Sudret, 2011]. $\boldsymbol{b} \in \mathbb{R}^{P+1}$ and $\boldsymbol{\psi}$ denote vectors which gather the coefficients and basis polynomials, respectively.

To determine the coefficients $b_{\boldsymbol{\alpha}}$ we project $f(X)$ onto the polynomial basis yielding

$$\langle f(X), \psi_{\boldsymbol{\alpha}}(X)\rangle = \langle \sum_{\boldsymbol{\alpha}\in\mathbb{N}^n} b_{\boldsymbol{\alpha}}\psi_{\boldsymbol{\alpha}}(X)(X), \psi_{\boldsymbol{\alpha}}(X)\rangle = b_{\boldsymbol{\alpha}}\langle \psi_{\boldsymbol{\alpha}}(X), \psi_{\boldsymbol{\alpha}}(X)\rangle, \tag{3.25}$$

due to the orthogonality of the polynomials and the linearity of the inner product. Assume that the polynomials are orthonormal then the coefficients are given as

$$b_{\boldsymbol{\alpha}} = \langle f(X), \psi_{\boldsymbol{\alpha}}(X)\rangle = \int_\Omega f(X(\omega))\psi_{\boldsymbol{\alpha}}(X(\omega))d\mathbb{P}(\omega) = \int_{\mathcal{X}} f(\boldsymbol{x})\psi_{\boldsymbol{\alpha}}(\boldsymbol{x})\pi(\boldsymbol{x})d\boldsymbol{x}. \tag{3.26}$$

With the coefficients at hand it is easy to approximate the mean and variance of $f(X)$ due to the orthogonality of the basis polynomials. The mean is simply $\mathbb{E}[f(X)] \approx b_{\mathbf{0}}$ and the variance $V[Y] \approx \sum_{\boldsymbol{\alpha}\in A(p)} b_{\boldsymbol{\alpha}}^2\langle \psi_{\boldsymbol{\alpha}}(\xi), \psi_{\boldsymbol{\alpha}}(\xi)\rangle - b_{\mathbf{0}}^2$

For non intrusive computation of the PCE coefficients two approaches are distinguished: the regression and the projection approach.

3.3.3. Estimation of the PCE coefficients via regression

For regression we formulate (3.25) in a vectorized version

$$\boldsymbol{b} = \mathbb{E}[\boldsymbol{\psi}(X)\boldsymbol{\psi}^T(X)]^{-1}\mathbb{E}[\boldsymbol{\psi}(X)f(X)], \tag{3.27}$$

where the matrix components are $\mathbb{E}[\boldsymbol{\psi}(X)\boldsymbol{\psi}^T(X)]_{\boldsymbol{\alpha}\boldsymbol{\beta}} = \langle \psi_{\boldsymbol{\alpha}}(X), \psi_{\boldsymbol{\beta}}(X)\rangle$. For given realizations $\{\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(M)}\}$ of X and corresponding evaluations $\boldsymbol{f} = \{f(\boldsymbol{x}^{(1)}), \ldots, f(\boldsymbol{x}^{(M)})\}$ we compute the empirical analogue of (3.27) by estimating the expectations via Monte Carlo. This yields the least squares approximation of the PCE coefficients

$$\hat{\boldsymbol{b}} = (\Psi^T\Psi)^{-1}\Psi F, \tag{3.28}$$

where the components of the matrix Ψ are $\Psi_{ij} = \psi_{\boldsymbol{\alpha}_j}(\boldsymbol{x}^{(i)})$ for $i = 1, \ldots, M$ and $j = 1, \ldots, P+1$. For details we refer to [Berveiller et al., 2006] and for least angle regression, an improvement w.r.t. to sparsity of the polynomial basis, to [Blatman and Sudret, 2011].

3.3.4. Estimation of the PCE coefficients via projection

For the projection approach the integral in (3.26) is in principle approximated via numerical integration or quadrature methods, i.e.

$$b_{\boldsymbol{\alpha}} = \int_{\mathcal{X}} f(\boldsymbol{x})\psi_{\boldsymbol{\alpha}}(\boldsymbol{x})\pi(\boldsymbol{x})d\boldsymbol{x} \approx \sum_{j=1}^{m} f(\boldsymbol{\lambda}_j)\psi_{\boldsymbol{\alpha}}(\boldsymbol{\lambda}_j)\nu_j, \tag{3.29}$$

with the quadrature nodes $\boldsymbol{\lambda}_j \in \mathcal{X}$ and weights $\nu_j \in \mathbb{R}, j = 1, \ldots, m$ determined with respect to $\pi(X)$. Quadrature methods differ by the way of determining the nodes and the corresponding weights. For a profound introduction to numerical integration we refer to [Quarteroni et al., 2010].

In this work we use the Smolyak sparse pseudo-spectral projection (PSP) method[3], using Sparse Grid numerical integration rules. We briefly introduce those methods and refer to [Bungartz and Griebel, 2004, Nobile et al., 2008, Constantine et al., 2012, Conrad and Marzouk, 2013] for more details. Further, we refer to the authors previous work, analyzing and applying the methodology to an industrial use case [John, 2016].

Full Grid numerical integration is in principle the tensorization of univariate deterministic Gauss-quadrature rules. For a given multi index $\boldsymbol{m} \in \mathbb{N}_0^n$ it yields for (3.26) the approximation

$$\hat{b}_{\boldsymbol{\alpha}} = \mathcal{Q}^{\boldsymbol{m}}(f\psi_{\boldsymbol{\alpha}}) = \sum_{\boldsymbol{j} \in I_{\boldsymbol{m}}} f(\boldsymbol{\lambda}_{\boldsymbol{j}})\psi_{\boldsymbol{\alpha}}(\boldsymbol{\lambda}_{\boldsymbol{j}})\nu_{\boldsymbol{j}}. \tag{3.30}$$

where the full tensor multi-index set is given by

$$I_{\boldsymbol{m}} = \{\boldsymbol{j} \in \mathbb{N}^n : 1 \leq j_i \leq n_{m_i} \text{ for } i = 1, \ldots, n\} \tag{3.31}$$

The n_{m_i} determine the number of nodes in each dimension and are defined via growth rules, such as the linear $n_m = 2m + 1$ or the exponential $n_m = 2^{(m+1)} - 1$ growth rules. However, Full Grid is usually not practicable, due to the curse of dimensions. Sparse Grid numerical integration, in principle, linearly combines a weighted selection of lower order Full Grid numerical integration schemes $\mathcal{Q}^{\boldsymbol{m}}$ of various resolutions $\boldsymbol{m}$ with the use of Smolyak's method $\mathcal{A}$ to partly circumvent the curse of dimension. I.e. for a given level $L \in \mathbb{N}_0$ the Sparse Grid approximation of the integral in (3.26) is

$$\hat{b}_{\boldsymbol{\alpha}} = \mathcal{A}(L, \mathcal{Q})(f\psi_{\boldsymbol{\alpha}}) = \sum_{\boldsymbol{m} \in M^s(L)} c(\boldsymbol{m}) \sum_{\boldsymbol{j} \in I_{\boldsymbol{m}}} f(\boldsymbol{\lambda}_{\boldsymbol{j}})\psi_{\boldsymbol{\alpha}}(\boldsymbol{\lambda}_{\boldsymbol{j}})\nu_{\boldsymbol{j}}. \tag{3.32}$$

where

$$M^s(L) = \{\boldsymbol{m} \in \mathbb{N}_0^n : L - n + 1 \leq |\boldsymbol{m}| \leq L\} \tag{3.33}$$

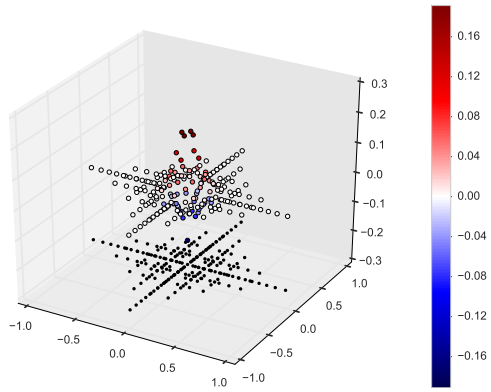
is the Smolyak multi-index set and

$$c(\boldsymbol{m}) = (-1)^{L-|\boldsymbol{m}|} \binom{n-1}{L-|\boldsymbol{m}|} \in \mathbb{R} \tag{3.34}$$

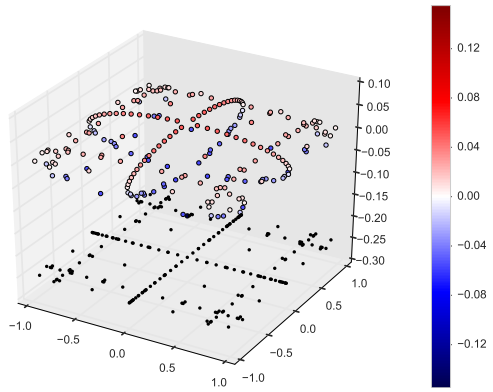
is the corresponding Smolyak combining coefficient [Conrad and Marzouk, 2013]. Simply computing the PCE coefficients $b_{\boldsymbol{\alpha}}$ yields the so-called non intrusive spectral projection (NISP) method.

[3] Introduced in [Constantine et al., 2012] as sparse pseudo-spectral approximation method (SPAM), it is in the ensuing literature better known as PSP [Conrad and Marzouk, 2013].

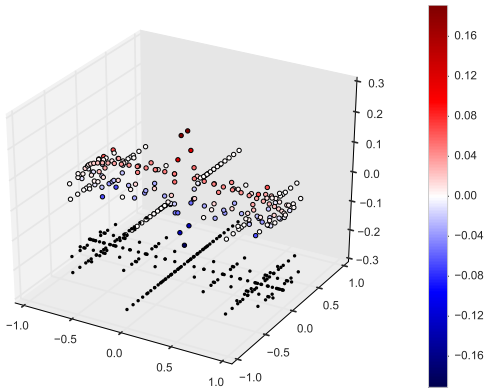
Example 1 (Illustration of Sparse grids in 2D)**.** Consider a two dimensional, independent random variable $X = [X_1, X_2] : \Omega \to \mathbb{R}^2$ with probability density π. Let $f(X) : \Omega \to \mathbb{R}$ be a function of X, such that $f(X) \in L^2(\omega)$. For the isotropic Smolyak Sparse Grid quadrature $\mathcal{A}(L, \mathcal{Q})$ the nodes $\boldsymbol{\lambda_j} \in \mathcal{X}$ and weights $\nu_j \in \mathbb{R}$, to approximate $\mathbb{E}[f(X)] = \int_\Omega f(X(\omega))dP(\omega)$ are visualized in Figure 3.1 for: (a) $X_1, X_2 \sim \mathcal{N}(0, 0.1)$, (b) $X_1, X_2 \sim \mathcal{U}[-1, 1]$ and (c) $X_1 \sim \mathcal{U}[-1, 1], X_2 \sim \mathcal{N}(0, 0.1)$. Depending on the distribution of X_i the univariate quadrature rules in dimension i are based on Legendre or Hermite polynomials for uniform or Gaussian distribution, respectively. Due to Smolyak's method (i.e. in particular due to Smolyak's combining coefficient), some of the weights have negative values. The level is set to $L = 4$ which results with the growth rule $n_m = 2^{(m+1)} - 1$ in a total of 221 nodes.



(a) Gaussian and Gaussian



(b) Uniform and Uniform



(c) Uniform and Gaussian

Figure 3.1.: Visualization of two dimensional, isotropic Smolyak Sparse Grid quadrature nodes $\boldsymbol{\lambda_j}$ (black dots) and weights $\boldsymbol{\nu_j}$ (colored circles) for level $L = 4$ (total number of 221 nodes) with X in (a), (b) and (c) specified as in Example 1. The figures are from [John, 2016].

NISP can be improved by the PSP method, which applies the Smolyak principle directly to univariate projection operators $\mathcal{P}$ (projecting on a univariate orthogonal polynomial basis) instead to quadrature operators. This naturally implies a coupling of the orthogonal polynomials used for projection with the orthogonal polynomials used for quadrature. Which automatically determines the quadrature methods to approximate the PCE coefficients in an optimal way, with the advantage of satisfying a discrete orthogonality property, leading to an error reduction (i.e. the reduction of internal aliasing errors [Conrad and Marzouk, 2013]) and allowing higher order polynomials. The PSP approximation or PSP surrogate of f reads

$$f(\boldsymbol{x}) \approx f^L(\boldsymbol{x}) = \mathcal{A}(L, \mathcal{P})(f)(\boldsymbol{x}) = \sum_{\boldsymbol{m} \in M^s(L)} c(\boldsymbol{m}) \sum_{\boldsymbol{i} \in I_{\boldsymbol{m}}} \hat{f}_{\boldsymbol{i}}\ \psi_{\boldsymbol{i}}(\boldsymbol{x}), \tag{3.35}$$

with coefficients obtained via Full Grid quadrature

$$\hat{f}_{\boldsymbol{i}} \approx \mathcal{Q}^{\boldsymbol{m}}(f\pi_{\boldsymbol{i}}) = \sum_{\boldsymbol{j} \in I_{\boldsymbol{m}}} f(\boldsymbol{\lambda_j})\psi_{\boldsymbol{i}}(\boldsymbol{\lambda_j})\nu_{\boldsymbol{j}}. \tag{3.36}$$

At this point we highlight that the choice of the full tensor quadrature $\mathcal{Q}^{\boldsymbol{m}}$, as tensor product of univariate Gauss-quadratures $\mathcal{Q}^{m_i}$, w.r.t. the orthonormal polynomials $\psi_{n_{m_i}+1}$ with $\deg(\psi_{n_{m_i}+1}) = n_{m_i}$, is one of the key points of PSP.

The accuracy of the surrogate f^L increases with the level L, but also the associated cost as the simulation model needs to be evaluated at an increasing number of nodes $\boldsymbol{\lambda_j}$.

Finally, the step to compute the PSP surrogate $\mathcal{M}^L$ are: First, evaluate $\mathcal{M}$ at the Sparse Grid nodes $\boldsymbol{\lambda}_j$ and second, compute the PCE coefficients component-wise for I and ω w.r.t. the N_t time points.

Remark. In this work we construct the PCE surrogate with a fixed Sparse Grid, based on the prior distribution used for Bayesian inference. This is done a-priori, i.e. before the inference procedure starts. Current research aims to build the surrogate iteratively during inference with active learning approaches. We refer to [Li and Marzouk, 2014, Sinsbeck and Nowak, 2017] for further reading.

3.3.5. Validation of PCE

As the PCE surrogate f^L of f is just an approximation, with errors due to truncation of the PCE and due to approximation of the PCE coefficients, validation of f^L is important for further processing. The Root Mean Square Error (RMSE) is one variant to compare f^L to f. For this purpose draw random samples $\boldsymbol{x}_i,\ i = 1, \ldots, N_{val}$ from X and evaluate the original model $f(\boldsymbol{x}_i),\ i = 1, \ldots, N_{val}$. Then the RMSE is defined as

$$\text{RMSE} = \left(\frac{1}{N_{val}} \sum_{i=1}^{N_{val}} |f(\boldsymbol{x}_i) - f^L(\boldsymbol{x}_i)|^2 \right)^{-\frac{1}{2}}. \tag{3.37}$$

For f with multiple output we compute the RMSE for each output component and scale it component-wise by the standard deviation of the validation set $\sigma_{val} = \text{std}(\{f(\boldsymbol{x}_i),\ i = 1, \ldots, N_{val}\})$, i.e. the scaled RMSE is

$$\text{sRMSE} = \text{RMSE}\,/\sigma_{val}. \tag{3.38}$$

Example 2. (PCE for forward UQ and validation) Consider the basic electric motor model (2.3), with $\mathcal{M}$ as a function of voltage V and load T. Define the input parameter distributions as $\pi(V \mid m_V, \sigma_V) = \mathcal{N}(m_V, \sigma_V^2)$ and $\pi(T \mid m_T, \sigma_T) = \mathcal{N}(m_T, \sigma_T^2)$, with hyper-parameters $m_V = 13.5,\ m_T = 2.5,\ \sigma_V = 0.7,\ \sigma_T = 0.2$. Then Figure 3.2 displays the forward propagation of uncertainty from V and T to output distributions of current I and angular velocity ω (Figure 2.2 above shows only the mean $\mathcal{M}(m_V, m_T)$). The output distributions are obtained by constructing a PCE $\mathcal{M}^L(V,T)$ of $\mathcal{M}(V,T)$ first, as detailed in Section 3.3.1, and then sampling from it. In particular we construct a PSP surrogate $\mathcal{M}^L(V,T)$ with level $L = 2$ and linear growth rule $n_m = 2m + 1$, leading to a two dimensional Sparse Grid with 17 nodes $\boldsymbol{\lambda_j}$. To obtain the output distributions we evaluate $\mathcal{M}^L(V,T)$ at 10^4 samples of (V,T). Already for this (admittedly toy) example the speed up with the surrogate is around factor 750 (51ms for the surrogate compared to 38s for the original model evaluation at 10^4 samples). For validation the sRMSE with $N_{val} = 100$ is displayed in Figure 3.3, respectively for each time step and output.

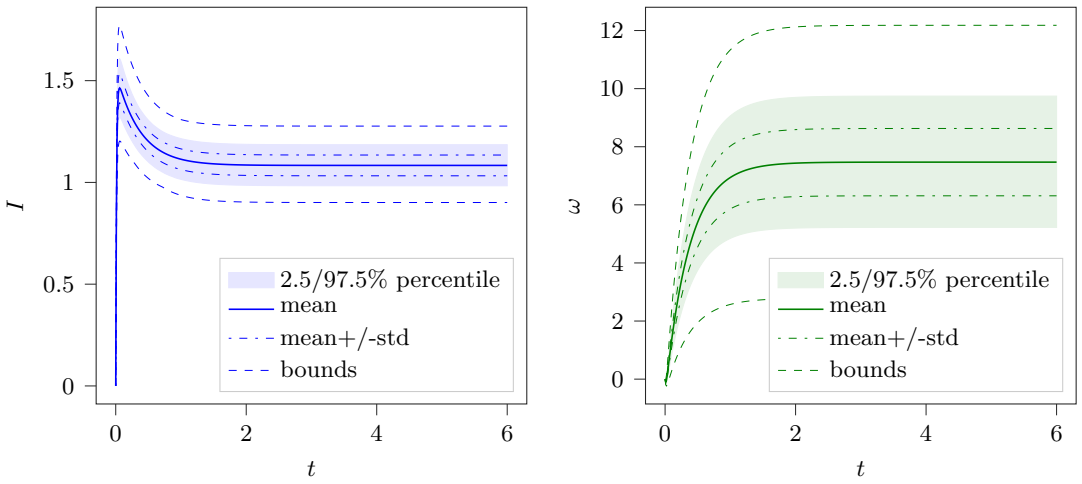


Figure 3.2.: Forward propagation of uncertainty for the basic electric motor model (2.3) from input probability distributions of voltage V and load T to output distributions of current I and angular velocity ω. The displayed overview on the output distributions are obtained by constructing a PCE $\mathcal{M}^L(V,T)$ of $\mathcal{M}(V,T)$ first and then sampling from $\mathcal{M}^L(V,T)$.

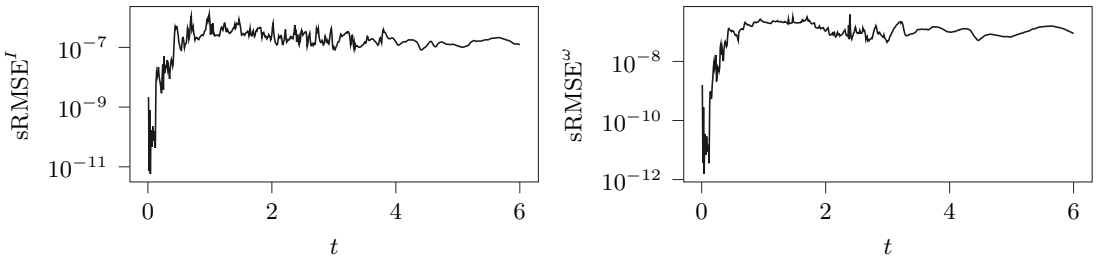


Figure 3.3.: The sRMSE for the PCE surrogate $\mathcal{M}^L(V,T)$ of $\mathcal{M}(V,T)$, respectively for each time step of current I and angular velocity ω.

3.3.6. Gaussian Processes

Gaussian Processes (GPs) are widely used in machine learning, e.g. for supervised learning tasks in regression and classification, and are also widely used in UQ as surrogates for simulation models. There exists extensive literature on GPs and GP regression (also well known as kriging, in particular in the geostatistics literature), thus we briefly introduce our notation for multi output GP regression and refer to the literature, e.g. [Rasmussen and Williams, 2006, Bonilla et al., 2007], for details. Note that the following text is based on and adapted from the authors own work [John et al., 2019a].

A GP is a stochastic process, i.e. a collection of random variables indexed by time or space, such that every finite collection of those random variables has a multivariate normal distribution. They can be used to describe the Bayesian a-priori uncertainty about a latent function and update this based on data.

Let $\boldsymbol{y} = f(t)$, for $t \in \mathbb{R}$ and $f : \mathbb{R} \to \mathbb{R}^d$. Consider the multi dimensional regression problem, where f is unknown and only a data matrix $D = [\boldsymbol{y}_i]_{i=1,\dots,N} \in \mathbb{R}^{d\times N}$, with $\boldsymbol{y}_i = f(t_i)$, at the corresponding discrete mesh $\Delta = \{t_1, \dots, t_N \,|\, t_n \in \mathbb{R}\}$, $t_m < t_n$ for $m < n$, is given. Denote by $vec(D) \in \mathbb{R}^{dN}$ the vectorization of matrix D. Further, let $A \otimes B = C$ denote the Kronecker product of $A \in \mathbb{R}^{n\times m}, B \in \mathbb{R}^{p\times q}$, then $C = [A_{ij}B]_{i=1,\dots,n,j=1,\dots,m} \in \mathbb{R}^{np\times mq}$.

GP regression assumes a prior for the unknown f by

$$P(f(t)) = \mathcal{GP}(f(t); m(t), k(t,t') \otimes V), \tag{3.39}$$

with prior mean function $m : \mathbb{R} \to \mathbb{R}^d$, a covariance kernel $k : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$ and $V \in \mathbb{R}^{d\times d}$ positive semi-definite. V correlates the d dimensional outputs of $f(t)$. The trivial choice for V is $V := I_d$, where $I_d \in \mathbb{R}^{d\times d}$ denotes the identity matrix. For the covariance kernel $k(t,t')$, define for two sets Δ, Δ' containing m and n elements, respectively, the $m \times n$ matrix $K_{\Delta\Delta'}$ with $(K_{\Delta\Delta'})_{i,j} = k(t_i, t'_j)$. Then the predictive posterior GP, conditional on the data D, is

$$P(f(t) \,|\Delta, D) = \mathcal{GP}(f(t); \underline{\mu}_{\mathcal{Y}}(t), \underline{k}_{\mathcal{Y}}(t,t')), \tag{3.40}$$

with

$$\underline{\mu}_{\mathcal{Y}}(t) = m(t) - (K_{t\Delta} \otimes V)G^{-1}vec(D), \tag{3.41}$$

$$\underline{k}_{\mathcal{Y}}(t,t') = K_{tt'} - (K_{t\Delta} \otimes V)G^{-1}(K_{\Delta t'} \otimes V), \tag{3.42}$$

where $G := K_{\Delta\Delta} \otimes V$. The t in $K_{t\Delta}$ symbolically stands for the set $\{t\}$. For the multi dimensional input case, with $t \in \mathbb{R}^n$ and $f : \mathbb{R}^n \to \mathbb{R}^d$, it requires a mean function $m : \mathbb{R}^n \to \mathbb{R}^d$ and a covariance kernel $k : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}^d$.

To actually compute the posterior GP a (parametric) mean function m, e.g. the trivial choice $m \equiv 0$, and a (parametric) covariance kernel k, e.g. the squared exponential kernel $k(t,t') = exp(|t-t'|^2/(2\lambda^2))$ with characteristic length scale $\lambda > 0$, need to be selected. As the computations are usually analytically intractable, approximation techniques are required to estimate the so-called hyper-parameters of the mean and covariance kernel. For details on various covariance kernels and their properties, model selection and hyper-parameter estimation see [Rasmussen and Williams, 2006]. There, relations to other techniques, such as SVMs, ANNs, splines and others, are discussed, too.

A useful property of GPs is their closeness under linear transformations [Bogachev, 1998]. Consequently, derivatives of GPs are again GPs, since differentiation is a linear operator [Solak et al., 2003]. For a covariance kernel $k(t,t')$, define $k^{\partial}(t,t') = \frac{\partial}{\partial t'}k(t,t')$, similarly ${}^{\partial}k(t,t') = \frac{\partial}{\partial t}k(t,t')$ and ${}^{\partial}k^{\partial}(t,t') = \frac{\partial^2}{\partial t \partial t'}k(t,t')$. Then, for $d = 1$, provided the derivatives exist,

$$P(\frac{d}{dt}f(t)) = \mathcal{GP}(f(t); \frac{d}{dt}m(t), {}^{\partial}k^{\partial}(t,t')). \tag{3.43}$$

Furthermore, $cov(f(t), \frac{d}{dt}f(t')) = k^{\partial}(t,t')$ and vice versa $cov(\frac{d}{dt}f(t), f(t')) = {}^{\partial}k(t,t')$.

3.4. Sensitivity Analysis

Sensitivity Analysis is "the study of how uncertainty in the output of a model (numerical or otherwise) can be apportioned to different sources of uncertainty in the model input" [Saltelli et al., 2008]. In particular one studies the effects of individual variations as well as the correlated effects of model parameters $x_i, i = 1, \ldots, n$ onto variations of a model $f(x_1, \ldots, x_n)$. Often one specific goal is to rank the input parameters x_i according to their importance or influence on the output of the model. This can then be further utilized for dimension reduction, i.e. to find an approximation $\hat{f}$ of f with fewer number of input parameters. In practice, uncertainty quantification and sensitivity analysis usually run in tandem.

Usually one distinguishes between local and global sensitivity analysis. Studying the dependence of f on local variations of the input parameters x_i around fixed values x_i^*, for instance with partial derivatives, we logically talk about local sensitivity analysis.

Global sensitivity analysis detects the sensitivity of f to variations over the whole parameter space $\mathcal{X}$. It can be seen as an average sensitivity. Global sensitivity analysis is established as a powerful approach for determining the important random input parameters that mainly drive the uncertainty of the model output. It can be distinguished in two classes: Regression-based methods and variance-based methods. In the following we briefly introduce variance-based methods, also called Analysis of Variance techniques (ANOVA), and the related Sobol' indices. For further details on global sensitivity analysis we refer to [Saltelli et al., 2008] and for details on the historic development of variance-based methods in particular to [Saltelli et al., 2008, Chapter 4]. ANOVA is a variance-based decomposition of a function f with n input parameters to determine which of the input parameters contribute most to variations of $f(x_1, \dots, x_n)$ and how they cooperate or compete. Based on the ANOVA decomposition Sobol' indices are a sensitivity measure describing which amount of the output variance is due to the uncertainties of a set of input parameters. They are very popular in sensitivity analysis since they provide accurate information for most physical and mathematical models and do not suppose linear or monotonic behavior of the model [Sobol, 2001, Sudret, 2008].

Consider the uncertain input parameters as a vector of independent random variables $\boldsymbol{X} = (X_1, \dots, X_n)$, with realization $\boldsymbol{x} = (x_1, \dots, x_n) = \boldsymbol{X}(\omega)$. Consequently, $Y = f(\boldsymbol{X})$ is a random variable as well (assume without loss of generality $f(\boldsymbol{x}) \in \mathbb{R}$). Let $\boldsymbol{X}_{\sim i} = (X_1, \dots, X_{i-1}, X_{i+1}, \dots, X_n)$ denote all random variables except X_i. With this we define the *first-order sensitivity measure*

$$S_i = \frac{V_{X_i}[\mathbb{E}_{\boldsymbol{X}_{\sim i}}[Y|X_i]]}{V[Y]}, \tag{3.44}$$

with $S_i \in [0, 1]$. This first-order sensitivity measure describes the main effect contribution of X_i on Y. Higher-order sensitivity measures can be obtained in a similar fashion by using variances conditioned on more than one random variable.

Sobol' decomposition, *Sobol' functional development* or *high-dimension model representation* (HDMR) expands a square integrable f into functional terms with increasing input dimension

$$\begin{aligned} f(x_1, \dots, x_n) = f_0 &+ \sum_{j=1}^{n} f_j(x_j) \\ &+ \sum_{1 \le j_1 < j_2 \le n} f_{j_1, j_2}(x_{j_1}, x_{j_2}) + \cdots \\ &+ \sum_{1 \le j_1 < \cdots < j_k \le n} f_{j_1, \dots, j_k}(x_{j_1}, \dots, x_{j_k}) + \cdots \\ &+ f_{1, \dots, n}(x_1, \dots, x_n), \end{aligned} \tag{3.45}$$

where f_0 is constant and each summand itself is square integrable [Saltelli et al., 2008]. It is not a series, since it has a finite number of summands $\sum_{j=0}^{n} \binom{n}{j} = 2^n$. This

decomposition allows to entirely identify contributions of single parameters $f_j(x_j)$ and contributions of parameter combinations $f_{j_1,\dots,j_k}(x_{j_1},\dots,x_{j_k})$ onto f. It is not unique, unless we assume that the mean of each summand, except of the constant f_0, over any of its independent variables, is zero, which was proven by Sobol in [Sobol, 2003]. As a consequence the summands are pairwise orthogonal and are defined as

$$f_0 = \mathbb{E}_{\boldsymbol{X}}[Y], \tag{3.46}$$
$$f_i(x_i) = \mathbb{E}_{\boldsymbol{X}_{\sim i}}[Y|X_i] - f_0, \tag{3.47}$$
$$f_{i,j}(x_i,x_j) = \mathbb{E}_{\boldsymbol{X}_{\sim i,j}}[Y|X_i,X_j] - f_i(x_i) - f_j(x_j) - f_0, \tag{3.48}$$
$$f_{i,j,k}(x_i,x_j,x_k) = \cdots \tag{3.49}$$

and so on. Finally, when all but $f_{1,\dots,n}(x_1,\dots,x_n)$ are obtained, the latter is defined via the identity (3.45).

Integrating the square of (3.45) yields the so-called ANOVA-HDMR and further normalizing by the total variance $V[Y]$ yields $2^n - 1$ terms, the so-called Sobol' indices, which give a full and unique sensitivity analysis of a model f with n-dimensional input.

Definition 6 (Sobol' indices)**.** *For an arbitrary set of indices $\{j_1,\dots,j_k\} \subseteq \{1,\dots,n\}$ and an integer $k \in \{1,\dots,n\}$ the relations*

$$S_{j_1,\dots,j_k} = \frac{V[f_{j_1,\dots,j_k}(X_{j_1},\dots,X_{j_k})]}{V[Y]}, \tag{3.50}$$

are called Sobol' indices, with $S_{j_1,\dots,j_k} \in [0,1]$. Each sensitivity index $S_{j_1,\dots,j_k}$ measures the relative impact of a set of input parameters $\{x_{j_1},\dots,x_{j_k}\}$ or respectively input random variables $\{X_{j_1},\dots,X_{j_k}\}$ onto the total variance $V[Y]$.

The first-order Sobol' indices

$$S_i = \frac{V_{X_i}[f_i(X_i)]}{V[Y]} = \frac{V_{X_i}[\mathbb{E}_{\boldsymbol{X}_{\sim i}}[Y|X_i]]}{V[Y]} \tag{3.51}$$

represent the influence of each parameter taken alone whereas the higher order indices $S_{j_1,\dots,j_k}$ account for possible mixed influence of various parameters. Two or more parameters interact, if their effect on the output cannot be expressed as a sum of their single effects [Saltelli et al., 2008].

In practical applications it is often sufficient to compute first-order, total order and sometimes second-order sensitivity indices to obtain a good, though non exhaustive characteristic of a model sensitivity [Sudret, 2008, Saltelli et al., 2008]. The classical computation of Sobol' indices requires Monte Carlo simulation to approximate the integrals, which fast becomes infeasible when computationally expensive simulation models are involved. However, here again surrogate models, such as the previously introduced Polynomial Chaos Expansions (PCE) or Gaussian processes (GP) help tremendously. They

do not only speed up Monte Carlo simulation, but also bring additional advantages, namely: Sobol' indices can be analytically derived from PCE [Sudret, 2008]. Whereas GPs straightforwardly allow to derive confidence intervals on the Sobol' indices, thus including the metamodel error. This is detailed in [Gratiet et al., 2016], where each technique is introduced with an focus on their strengths and limitations in the context of global sensitivity analysis. There, numerical comparison of PCE and GP based sensitivity analysis shows similar performance.

Example 3 (Forward UQ and first-order Sobol' indices)**.** With the same setting as in Example 2, where Figure 3.2 displays the forward propagation of uncertainty from $X = (V, T)$ to output distributions of current I and angular velocity ω, here, corresponding to this variations, the first order Sobol' indices $S_1^{(\cdot)}$ for I and ω w.r.t. the input random variables $X = (V, T)$ are displayed in Figure 3.4.

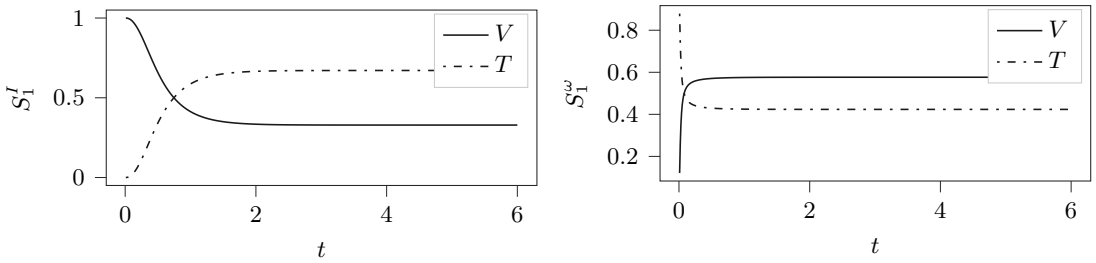


Figure 3.4.: First order Sobol' indices $S_1^{(\cdot)}$ for current I and angular velocity ω of the basic electric motor model (2.3).

3.5. Stability analysis of Bayesian inference with surrogate models

The simulation model $\mathcal{M}$ is an approximation of the true underlying process and additionally $\mathcal{M}$ is often further approximated by a surrogate for computational purposes. This leads to an approximation of the posterior measure $\mu^{\mathcal{Y}}$. Furthermore, generating samples that approximate the posterior distribution, e.g. with MCMC or ABC methods, introduces additional errors. For stability analysis we study the effect of small changes in the forward model, respectively in the least-squares potential Φ, on changes in the posterior distribution approximation, which need to be small as well to obtain stability. This analysis can be used to translate errors in the forward problem into estimates on errors in the posterior distribution.

The following analysis assumes fixed data $\mathcal{Y}$ and concentrates solely on changes in the forward model. For well posedness w.r.t to changes in the data, we refer to [Dashti and Stuart, 2017, Ch. 4.1].

Let $\mathcal{M}^L$ denote an approximation of $\mathcal{M}$, where L determines the approximation quality. Respectively, let Φ^L denote the approximation of the least-squares potential Φ. Further denote by $\tilde{\mu}_L^{\mathcal{Y}}$ the approximation of the posterior measure $\mu^{\mathcal{Y}}$ and by $\tilde{\pi}_L(\boldsymbol{x}|\mathcal{Y})$ the corresponding Lebesgue density. The following derivation holds for general approximations $\mathcal{M}^L$ that converge with $L \to \infty$ to $\mathcal{M}$. In particular for the (truncated) PCE, where $\mathcal{M}^L$ denotes the PCE surrogate of $\mathcal{M}$ for a level $L \in \mathbb{N}_0$.

Assumptions 1. *Assume that $\Phi \in C(\mathbb{R}^n;\mathbb{R})$ and there are functions $M_i : \mathbb{R}_+ \to \mathbb{R}_+$, $i = 1, 2$ independent of L and monotonic non-decreasing separately in each argument, and with M_2 strictly positive, such that for all $\boldsymbol{x} \in \mathbb{R}^n$*

$$\Phi(\boldsymbol{x}) \geq -M_1(\|\boldsymbol{x}\|) \tag{3.52}$$

$$\Phi^L(\boldsymbol{x}) \geq -M_1(\|\boldsymbol{x}\|) \tag{3.53}$$

$$\|\Phi(\boldsymbol{x}) - \Phi^L(\boldsymbol{x})\| \leq M_2(\|\boldsymbol{x}\|)\psi(L)\,, \tag{3.54}$$

where $\psi(L) \to 0$ as $L \to \infty$. Further assume that $\mu_0(\mathbb{R}^n \cap B) > 0$ for some bounded set $B \subset \mathbb{R}^n$ and

$$\exp(M_1(\|\boldsymbol{x}\|))(1 + M_2(\|\boldsymbol{x}\|)^2) \in L^1_{\mu_0}\,. \tag{3.55}$$

Definition 7. *(Hellinger distance) For two probability measures μ and μ' on a separable Banach space, which are both absolutely continuous with respect to a common reference measure ν, the Hellinger distance is defined by*

$$\mathrm{d}_{\mathrm{Hell}}\left(\mu, \mu'\right) = \sqrt{\frac{1}{2}\left(\frac{d\mu}{d\nu} - \frac{d\mu'}{d\nu}\right)^2 d\nu}. \tag{3.56}$$

Theorem 2. *[Dashti and Stuart, 2017, Theorem 4.9] Let Assumptions 1 hold. Then there is a $C > 0$ such that, for all L sufficiently large,*

$$\mathrm{d}_{\mathrm{Hell}}\left(\mu^{\mathcal{Y}}, \tilde{\mu}_L^{\mathcal{Y}}\right) \leq C\psi(L). \tag{3.57}$$

For functions φ with $\varphi \in L^2_{\mu^{\mathcal{Y}}}$ and $\varphi \in L^2_{\tilde{\mu}_L^{\mathcal{Y}}}$, uniformly w.r.t. L, the closeness of the Hellinger metric implies closeness of expectations of φ:

$$\left|E^{\mu^{\mathcal{Y}}}[\varphi(\boldsymbol{x})] - \mathbb{E}^{\tilde{\mu}_L^{\mathcal{Y}}}[\varphi(\boldsymbol{x})]\right| \leq C\,\mathrm{d}_{\mathrm{Hell}}\left(\mu^{\mathcal{Y}}, \tilde{\mu}_L^{\mathcal{Y}}\right) \leq C\psi(L). \tag{3.58}$$

Theorem 2 enables the translation of errors arising from approximation of the forward problem into errors in the Bayesian solution of the inverse problem. Furthermore, the errors in the forward and inverse problems scale the same way with respect to L. This result applies to the case, where $\mathcal{M}$ is an approximation of the true process, but also to the case where $\mathcal{M}$ itself is approximated by a surrogate model $\mathcal{M}^L$.

Stability analysis of ABC and MCMC with surrogate models

Theorem 2 covers the approximation of the posterior measure due to an approximation of the forward map. But in practice we additionally use sampling methods to generate samples of the approximated posterior measure. Assume that the goal of computation is to compute the mean of a quantity of interest φ w.r.t. the posterior $\mu^{\mathcal{Y}}$ with Lebesgue density $\pi(\boldsymbol{x}|\mathcal{Y})$. Expectations of this kind allow the study of many relevant properties of the posterior, for instance the posterior mean when $\varphi(\boldsymbol{x}) = x_i$ with $i = 1, \dots, n$ or posterior second moments when $\varphi(\boldsymbol{x}) = x_i x_j$ with $i, j = 1, \dots, n$.

Given the summary statistic $S : \mathbb{R}^{k \times N_t} \to \mathbb{R}^d$ with $d \leq kN_t$, the inference using ABC with summary statistics, see Algorithm 3, is based on $s^\dagger = S(\mathcal{Y})$ and constructs samples $\tilde{\boldsymbol{x}}_j^{(\delta)}$ based on the polynomial chaos surrogate. Assuming S to be a sufficient statistic, i.e.

$$\mathbb{E}^{\tilde{\mu}^{\mathcal{Y}}}[\varphi(\boldsymbol{x})|\mathcal{Y}] = \mathbb{E}^{\tilde{\mu}^{\mathcal{Y}}}[\varphi(\boldsymbol{x})|s^\dagger] \tag{3.59}$$

for the surrogate-based model, we are interested in the mean-square-error

$$\mathbb{E}^{\mu_0}\left[\left(\mathbb{E}^{\mu^{\mathcal{Y}}}[\varphi(\boldsymbol{x})|\mathcal{Y}] - \frac{1}{m}\sum_{j=1}^{m}\varphi(\tilde{\boldsymbol{x}}_j^{(\delta)})\right)^2\right]. \tag{3.60}$$

Theorem 3. *[John et al., 2020] Let the quantity of interest $\varphi : \mathbb{R}^n \to \mathbb{R}$ with $\varphi \in L^1_{\mu_0}$. In addition let $\varphi \in L^2_{\mu^{\mathcal{Y}}}$ and $\varphi \in L^2_{\tilde{\mu}_L^{\mathcal{Y}}}$, uniformly in L. Further let Assumptions 1 hold. Then, the mean square error converges to 0 for $L, m \to \infty$ and $\delta \to 0$, i.e.*

$$\mathbb{E}^{\mu_0}\left[\left(\mathbb{E}^{\mu^{\mathcal{Y}}}[\varphi(\boldsymbol{x})|\mathcal{Y}] - \frac{1}{m}\sum_{j=1}^{m}\varphi(\tilde{\boldsymbol{x}}_j^{(\delta)})\right)^2\right] \to 0\,, \quad m, L \to \infty,\ \delta \to 0\,. \tag{3.61}$$

Proof. The triangle inequality gives the separate estimation of the individual errors

$$\begin{aligned}
&\mathbb{E}^{\mu_0}\left[\left(\mathbb{E}^{\mu^{\mathcal{Y}}}[\varphi(\boldsymbol{x})|\mathcal{Y}] - \frac{1}{m}\sum_{j=1}^{m}\varphi(\tilde{\boldsymbol{x}}_j^{(\delta)})\right)^2\right]^{1/2} \qquad (3.62)\\
&\leq \left|E^{\mu^{\mathcal{Y}}}[\varphi(\boldsymbol{x})|\mathcal{Y}] - \mathbb{E}^{\tilde{\mu}_L^{\mathcal{Y}}}[\varphi(\boldsymbol{x})|\mathcal{Y}]\right| + \mathbb{E}^{\mu_0}\left[\left(\mathbb{E}^{\tilde{\mu}_L^{\mathcal{Y}}}[\varphi(\boldsymbol{x})|\mathcal{Y}] - \frac{1}{m}\sum_{j=1}^{m}\varphi(\tilde{\boldsymbol{x}}_j^{(\delta)})\right)^2\right]^{1/2}\\
&=: I_1 + I_2.
\end{aligned}$$

The assumptions on the least-squares potential ensure the closeness of the two measures $\mu^{\mathcal{Y}}$ and $\tilde{\mu}_L^{\mathcal{Y}}$ in the Hellinger distance, see Theorem 2, which implies

$$I_1 \leq C\psi(L)\,. \tag{3.63}$$

The second summand I_2 corresponds to the ABC error, which converges to 0 as $m \to \infty$ and $\delta \to 0$, see [Barber et al., 2015, Proposition 3.1]. □

Analogously, for inference using the MH-MCMC algorithm 1, which constructs samples $\tilde{\boldsymbol{x}}_j$ used to approximate

$$\mathbb{E}^{\tilde{\mu}_L^{\mathcal{Y}}}[\varphi(\boldsymbol{x})|\mathcal{Y}] \approx \frac{1}{m}\sum_{j=1}^{m}\varphi(\tilde{\boldsymbol{x}}_j), \tag{3.64}$$

the second summand I_2 converges to 0 as $m \to \infty$, see [Kaipio and Somersalo, 2005, Proposition 3.11] (I.e. the MCMC transition kernel needs to be invariant, aperiodic and irreducibel in order to imply the ergodicity property of MCMC samples). Further, see [Dashti and Stuart, 2017] for general measure-preserving dynamics on the infinite-dimensional space, including MCMC and SMC.

3.6. Related work on inverse problems

Gradient matching techniques are notable methods in the context of ODE parameter estimation. The general idea is to approximate the data by a differentiable approximator (e.g. Gaussian processes, see [Solak et al., 2003]), which is then used to directly match the derivative of the data approximation to the ODE function. The advantage is that no numerical solver for the ODE is required. For a kernel based approach see [Niu et al., 2016] and the references therein (Side note: For connections between GPs and kernel methods, see [Kanagawa et al., 2018].). References for P-spline and parallel tempering based approaches can be found in [Niu et al., 2016]. For GP based approaches the controversy [Macdonald et al., 2015] shows that the model suggested in [Barber and Wang, 2014] suffers from an inherent identifiability problem, which is not the case for the superior models in [Calderhead et al., 2008, Dondelinger et al., 2013]. The latter work is further extended for scalability and efficiency in [Gorbach et al., 2017, Wenk et al., 2019b, Wenk et al., 2019a]. Also, extensions to SDEs exist [Abbati et al., 2019]. However, in general gradient matching methods are not suited for black box forward models, as they require direct access to the ODEs.

Invertible Neuronal Networks (INNs) are another promising approach to inverse problems, see [Ardizzone et al., 2019] and the references therein. INNs are designed to learn the forward model, from parameters to output, and use additional latent output variables to encode information that would otherwise be lost. Due to construction INNs implicitly learn a model of the corresponding inverse problem. This can be used to map a measurement to the parameter posterior distribution by sampling from the latent variables. However, as they only learn from the forward map, model discrepancy is not considered so far.

Chapter 4.

Discrepancy modeling

> The fact that the polynomial is an approximation does not necessarily detract from its usefulness because all models are approximations. Essentially, all models are wrong, but some are useful. However, the approximate nature of the model must always be borne in mind.
>
> George Box and Norman Draper in [Box and Draper, 1987, p. 424]

As already motivated in Section 1.3 it is crucial to quantify model discrepancy in inverse problems. To succed in this it is important to be aware of and additionally quantify the effects of further uncertainty sources, as they otherwise could undermine the reliability of results. This chapter addresses the inverse problem in Definition 4, where one seeks model parameters $\boldsymbol{x} \in \mathcal{X}$ such that $\boldsymbol{y} = \mathcal{M}(\boldsymbol{x})$ for noisy measurements $\boldsymbol{y} \in \mathbb{R}^{2\times N_t}$ and simulation model $\mathcal{M}$. In general, this equality posseses either none, one or multiple solutions in $\boldsymbol{x}$. Thus, a proper indentification and quantification of possible solutions providing additional information on their uncertainty is of particular interest for most engineering tasks. Furthermore, some applications require realistic calibration of physically interpretable parameters $\boldsymbol{x}$ togheter with an assessment of model quality and an reliable model discrepancy quantification. To be precise we define:

Definition 8 (Model discrepancy)**.** *The term model discrepancy $\boldsymbol{\delta} = \mathcal{M}(\boldsymbol{x}) - \boldsymbol{\eta}$ denotes the difference between the true system $\boldsymbol{\eta}$ and the simulation model $\mathcal{M}$. The true underlying model discrepancy is $\boldsymbol{\delta}^\dagger = \mathcal{M}(\boldsymbol{x}^\dagger) - \boldsymbol{\eta}$, with the true physical parameters $\boldsymbol{x}^\dagger$. It comprises the structural uncertainty that might be due to missing physics in the formulation of the simulation model and also numerical errors.*

The joint identification of model discrepancy $\boldsymbol{\delta}$ and simulation model parameters $\boldsymbol{x}$ is one of the most fundamental challenges in solving inverse problems as both are usually unknown. It is often denoted by:

Definition 9 (Identification problem)**.** *The identification problem is the dilema of distinguishing between effects of the model parameters $\boldsymbol{x}$, the model discrepancy $\boldsymbol{\delta}$ and the observation noise $\boldsymbol{\varepsilon}$. In principle for every $\boldsymbol{x}$ a $\boldsymbol{\delta}$ and $\boldsymbol{\varepsilon}$ exist such that $\boldsymbol{y} = \mathcal{M}(\boldsymbol{x}) + \boldsymbol{\delta} + \boldsymbol{\varepsilon}$ holds. Neglecting $\boldsymbol{\varepsilon}$ for a moment, then $\boldsymbol{\delta} := \boldsymbol{y} - \mathcal{M}(\boldsymbol{x})$ would satisfy the previous equation for each $\boldsymbol{x} \in \mathcal{X}$.*

This chapter addresses this problem and showcases it with the electric motor application by adressing following questions: How good does the model explain the data? Can it be improved by modeling the model discrepancy? Does this help to infer model parameters $\boldsymbol{x}$ that are not only mathematically optimal, but correspond to the true underlying physical parameter values? And what possibilities are there to weaken the problem inherent identifiability issue?

The remainder of this chapter is structured as follows. Section 4.1 gives an overview on related work and Section 4.2 remarks on numerical approximation error and probabilistic numerics. Following, we introduce our Bayesian discrepancy modeling framework in Section 4.3, highlight the specific challenges in Section 4.3.1, detail on Gaussian process approximation in Section 4.3.2 and on orthogonal function approximation with iterative model complexity determination in Section 4.3.3. Section 4.4 describes the improvement of sampling with surrogate-based gradients. Detailed numerical results of the methods applied to synthetic and test bench data are presented in Section 4.5.

4.1. Related work

Addressing model discrepancy in inverse problems requires to incorporate the underlying physics and the associated simulation model, thus many attempts in dealing with model discrepancy are hidden in domain specific publications. Consequently a comprehensive literature review is difficult. In the following, we first give an overview on some highly relevant publications for this thesis and second open the scope for publications that deal with the problem in another way.

The Kennedy and O'Hagan (KO) framework [Kennedy and O'Hagan, 2001] is one of the first attempts to model and explicitly take account of all uncertainty sources arising in the calibration of computer models. In particular, model discrepancy is considered by an additional term δ in the Bayesian formulation of the inverse problem, i.e.

$$\boldsymbol{\eta} = \mathcal{M}(\boldsymbol{x}) + \boldsymbol{\delta}, \tag{4.1}$$

$$\boldsymbol{y}_i = \boldsymbol{\eta}_i + \boldsymbol{e}_i, \quad i \in \mathbb{N}, \tag{4.2}$$

where $\boldsymbol{\eta}$ denotes the true physical process and $\boldsymbol{y}$ are a collection of noisy observations of $\boldsymbol{\eta}_i = \eta(t_i)$ at time points t_i, for some observation error $\boldsymbol{e}_i$. Kennedy and O'Hagan use Gaussian Processes (GPs) to model $\boldsymbol{\delta}$ and, as well, the simulator $\mathcal{M}$. One of their conclusions is that the estimated parameters depend on the given data and defined error structure. Those estimates do not necessarily correspond to the true physical values. Often the computer model explains the data better for other values and restriction of parameters to the true physical value might lead to worse results. Derivatives of [Kennedy and O'Hagan, 2001] are among others [Higdon et al., 2004, Kennedy et al., 2006, Conti et al., 2009]. A lot of the works referring to the KO framework only use the approxima-

tion of the original simulator by a GP, which then reflects the approximation uncertainty, but do not consider model discrepancy in order to avoid identification problems.

Following [Kennedy and O'Hagan, 2001], Arendt et al. [Arendt et al., 2012a, Arendt et al., 2012b] suggest a modular Bayesian approach to solve (4.1). First, the hyper-parameters of the GP emulating the simulator and the hyper-parameters of the model discrepancy GP are approximated by maximum likelihood estimates consecutively. Second, the posterior distributions of the remaining parameters are approximated, conditioned on the data and the hyper-parameter estimates. Discussion of the identification problem with illustrative examples shows that a separation of effects is sometimes possible under mild assumptions, e.g. smoothness of the model discrepancy, but also that it is not possible in other cases. [Arendt et al., 2012b] presents an approach to improve identifiability by using multiple responses and representing correlation between responses. Another work using multiple responses is [Paulo et al., 2012].

Simoen et al. [Simoen et al., 2013] also follow the KO approach, using the term prediction error model as synonym for the joint statistical model of discrepancy and noise (also termed likelihood). As most practical applications lack information regarding prediction error characteristics, they suggest Bayesian model class selection to select the most probable prediction error model from several alternative model classes, according to the available data. Further, they observe that the prediction error model has an important effect on the parameters posterior distribution.

Brynjarsdóttir and O'Hagan [Brynjarsdóttir and O'Hagan, 2014] state that with the KO framework, in order to infer physical parameters and model discrepancy simultaneously, a sufficient prior distribution for at least one of those must be given. An example shows that a constrained GP prior for the model discrepancy, including best and most realistic prior information, yields good results for interpolation and learning about physical parameters, but still seems to be bad for extrapolation. Hence, simply introducing model discrepancy with weak prior information is not enough.

Alongside the Bayesian approach to model discrepancy there exists also a second major thrust of research, the large sample frequentist approach to model uncertainty, see the introduction of [Plumlee, 2019] and the references therein. Among those, Tuo and Wu [Tuo and Wu, 2015, Tuo and Wu, 2016] criticize the KO approach and show that the choice of the model discrepancy prior has a permanent influence onto the parameter posterior distribution even in the large data limit. Consequently, one needs to be careful with specific prior information. They further introduce the so called L^2-projected calibration (a frequentist method), where the optimal parameter is defined by

$$\boldsymbol{x}_{\text{opt}} = \arg\min_{\boldsymbol{x}\in\mathcal{X}} \|\boldsymbol{\eta} - \mathcal{M}(\boldsymbol{x})\|_{L^2}^2 = \arg\min_{\boldsymbol{x}\in\mathcal{X}} \int |\eta(t) - \mathcal{M}(t,\boldsymbol{x})|^2 dt. \tag{4.3}$$

$\boldsymbol{x}_{\text{opt}}$ minimizes the L^2-norm of the model discrepancy $\boldsymbol{\delta} = \boldsymbol{\eta} - \mathcal{M}(\boldsymbol{x})$ which forces the model $\mathcal{M}$ to explain most of the variations of $\boldsymbol{\eta}$. In practice $\boldsymbol{\eta}$ is not available and is replaced by an estimation $\hat{\boldsymbol{\eta}}$ based on an approximation of the noisy data $\boldsymbol{y}$ (e.g.

with kernel ridge regression or GP regression). The L^2-projected calibration avoids the identifiability problem and has nice asymptotic properties. However, for UQ additional procedures are required (e.g. bootstrapping, see [Wong et al., 2017]). A further drawback is that $\boldsymbol{x}_{\text{opt}}$ not necessarily corresponds to the true physical parameters.

A bunch of works follow this L^2-projected calibration, develop it further and partly combine it with the Bayesian KO approach. Those are among others [Plumlee, 2017, Gu and Wang, 2018, Plumlee, 2019, Tuo, 2019, Xie and Xu, 2020], detailed in the following.

Plumlee [Plumlee, 2017] generalizes the L^2-norm and combines this with the Bayesian KO approach to mitigate identifiability problems in the latter. As a consequence of the combination, a GP prior distribution of the model discrepancy is defined that is orthogonal to the gradient of the model. The examples show decent results, but at the cost of additional computational effort and the availability of a gradient. This adds to the already high costs for modeling the model discrepancy with GPs for large number of observations N_t. I.e. each single sample requires the inversion of an $N_t \times N_t$-covariance matrix, which results in numerical costs that scale with $\mathcal{O}(N_t^3)$. [Gu and Wang, 2018] also introduce a modified GP prior, referred to as the scaled GP, which is directly applied to the discrepancy function. This work focuses on prediction and not on interpretability of the parameters. The method proposed in [Plumlee, 2019] produces a conservative confidence set on the parameters that includes the best parameter with a desired probability. Additionally, it is consistent in that it excludes suboptimal parameters in large sample environments. The set is conservative and consistent, two properties whose coincidence is not present in previously existing Bayesian or large sample frequentist methods. The caveat is the requirement of a user specified bound on the norm of the discrepancy. Further, discrepancy correction is not explicitly explored in this paper. In [Tuo, 2019] a frequentist method, called the projected kernel calibration method, is proposed that has a natural Bayesian version. They show that the inconsistency problem of the KO approach can be rectified by a simple modification of the kernel matrix. In the recent work [Xie and Xu, 2020] an approach called Bayesian projected calibration (regarded as the Bayesian version of the L^2-projected calibration) is proposed. Following [Tuo and Wu, 2015, Tuo and Wu, 2016, Wong et al., 2017, Plumlee, 2017, Tuo, 2019], they add theoretical guarantees. We refer the interested reader to [Xie and Xu, 2020] as a starting point for further reading on L^2-projected calibration and its developments.

Nagel et al. [Nagel et al., 2017] (recently published in [Nagel et al., 2020]) follow the KO approach and address the point of high computational costs with GPs. They employ a Principal Component Analysis (PCA) w.r.t. the simulation model for identification of important data points in order to reduce the data and thus computational costs. They further model the model discrepancy term by a low degree polynomial expansion, assuming smoothness for the true underlying model discrepancy. This, in combination with a zero-mean GP with an exponential kernel for the measurement noise, leads to a decent data-model fit. Conclusions about the quality of the inferred model parameters are not possible, due to the lack of reference values.

Another approach in [Goulet and Smith, 2013] and the references therein is called error-domain model-falsification (EDMF) motivated for civil engineering applications. EDMF avoids the definition of correlations (they use the term uncertainty dependencies) and is instead based on the concept of falsification. Falsification is the philosophical perspective that scientific models cannot be fully validated by data, they can only be falsified, see [Popper, 2005]. Comparison of residual minimization, Bayesian inference and EDMF methods yields following conclusions. In presence of aleatoric and systematic errors one needs to be careful with independence assumptions in Bayesian inference, as the posterior distribution might be biased, which is in line with the findings in [Simoen et al., 2013] and [Brynjarsdóttir and O'Hagan, 2014]. The same holds for residual minimization, where the point estimates are biased. In an example they show that EDMF can, given sufficient measurements, identify when initial assumptions are flawed by falsifying candidate models. However a result could be that all candidate models are falsified. In the examples the EDMF results were able to infer the correct parameters, but the parameter estimates were very conservative (desirable for civil engineering applications). Additional comparison of Bayesian inference and EDMF w.r.t. robustness and extrapolation can be found in [Pasquier and Smith, 2015].

Other approaches are intrusive or embedded model corrections. In [Parish and Duraisamy, 2016] a machine learning approach with full-field inversion, requiring inverse modeling to address model form-error, is suggested to aid the creation of improved closure models for computational physics applications. However, as an intrusive approach it is not suitable for black-box models. In the context of chemical kinetics [Morrison et al., 2018] directly embed a discrepancy operator in the dynamical system. It respects certain physical constraints such as conservation laws and requires the definition and inference of several additional hierarchically modeled parameters, adding to the burden of analysis. [Sargsyan et al., 2019] follow this embedded approach with PCEs for internal model correction, which might be beneficial for extrapolation outside of testing range. A further alternative is presented in [Bhat et al., 2017], where an embedded, dynamic discrepancy is added. Modeling it by a GP turns the governing ODEs or PDEs into a system of stochastic differential equations. [Bruder and Koutsourelakis, 2018] also opt for opening black-box forward models to quantify model uncertainty in a physically meaningful manner. They formulate an undirected probabilistic model, which recasts the solution of both forward and inverse problems as probabilistic inference tasks.

Recently, more Machine Learning driven intrusive approaches evolved. E.g. in [Han et al., 2018] a deep learning based approach is introduced that can handle general high-dimensional parabolic PDEs. The PDEs are reformulated as backward stochastic differential equations, where the gradient of the unknown is approximated by neural networks. In [Raissi et al., 2019] neural networks are trained to solve supervised learning tasks while respecting any given laws of physics described by general nonlinear PDEs. [Girolami et al., 2019] also systematically incorporate data into the finite element method (FEM) solution of PDEs in the face of model misspecification using GPs. Recently,

following [Chen et al., 2018], Rackauckas et al. [Rackauckas et al., 2020] proposed latent (Neural) ODEs, where the ODE function is learned from data, e.g. with neural networks or any other universal approximator. If the ODE function is partly known the missing parts can be learned from data. This process can be seen as machine learning based model discovery and is still current research. However, in general, simultaneous learning of parameters from known ODE function parts and learning of unknown ODE function parts yields again identification problems.

Finally, albeit not explicitly considering model discrepancy, Multilevel or Multifidelity Monte Carlo methods need to be mentioned for the case where multiple computational models with varying evaluation cost and fidelity are available. Those methods combine computationally (often too) expensive high-fidelity models with lower-fidelity models that are less accurate but computationally cheaper. I.e. this can be seen as a correction of the lower-fidelity model for the task at hand, such as optimization, inference and uncertainty quantification, with only minimal amount of high-fidelity model evaluations to establish accuracy. See [Giles, 2015] for Multilevel Monte Carlo and [Peherstorfer et al., 2018] for a survey on Multifidelity Monte Carlo methods. This approach cannot only be applied on the simulation model level, but also for averaging of several competing (physically motivated) discrepancy models. In [Edeling et al., 2014b] and [Edeling et al., 2014a] additive measurement noise and multiplicative model discrepancy term are considered for turbulence closure models, the dominant error source in most Reynolds-Averaged Navier–Stokes simulations. In [Edeling et al., 2014a] they develop a method called Bayesian Model-Scenario Averaging, where several turbulence closure models are averaged. They state that this approach can be successful when a large amount of data is available and when model parameters are fixed.

4.2. Remark on numerical approximation error

Numerical approximation error is the inherent, epistemic uncertainty associated with approximating abstract mathematical models with numerical methods. For ODEs this is due to the finite-dimensional approximation of an unknown and implicitly defined function. In some applications it might be important to explicitly account for the uncertainty associated with the numerical method, in order to relate it to other sources of uncertainty, e.g. measurement noise or model discrepancy due to missing modeling [Conrad et al., 2017]. Probabilistic numerics (PN) aims at providing a growing toolbox of methods that address these issues. PN methods recast numerical approximation as an inference problem and return a distribution (likelihood) over possible solutions. This distribution represents the inherent numerical uncertainty in contrast to a sole point estimate in classical deterministic solvers. For an introduction to PN we refer to [Hennig et al., 2015]. For PN w.r.t. ODEs see [Schober et al., 2019], where classical methods, such as Runge–Kutta methods, are formulated in a probabilistic way and interpreted as Gaus-

sian ODE filters. Another contribution to the growing class of PN methods is GOODE (acronym for Gaussian Off-the-shelf Ordinary Differential Equation solver), the first PN method for nonlinear 2-point Boundary Value Problems (BVPs) [John et al., 2019a]. GOODE iteratively approximates nonlinear BVPs by a sequence of linear BVPs, which are solved with GP regression, respectively. GOODE also solves Initial Value Problems (IVPs), since IVPs can be recast as BVPs. Figure 4.1 presents a GOODE solution for the trajectory of current I and angular velocity ω for the basic electric motor model (2.3), solved with a squared exponential kernel and 201 equidistant time steps in the time interval $[0, 6]$. The resulting GP represents with its distribution the uncertainty due to numerical approximation with the finite mesh of 201 time steps. Actually, the standard deviation is an upper bound on the local numerical approximation error, see [John et al., 2019a] for details.

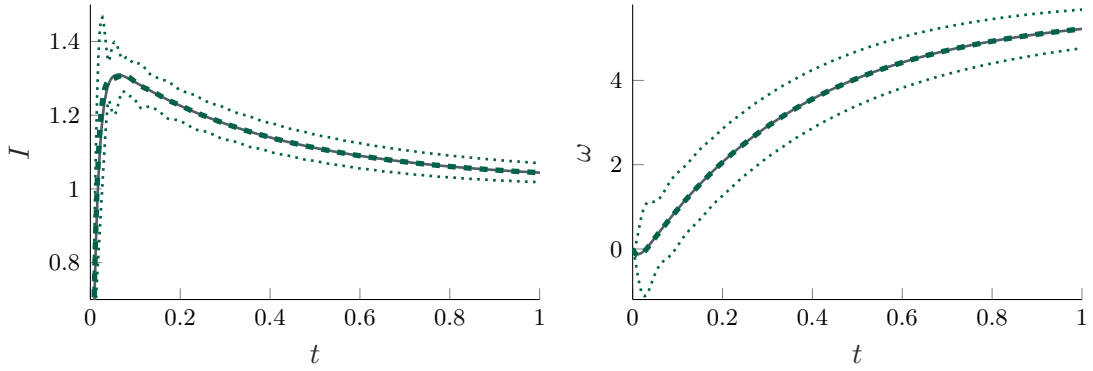


Figure 4.1.: GOODE solution for the trajectory of current I and angular velocity ω for the basic electric motor model (2.3), solved with a squared exponential kernel and 201 equidistant time steps in the time interval $[0, 6]$. For better visualization only the time frame $[0, 1]$ is shown. Legend: reference solution with Matlab's ode45 (solid, gray); GOODE mean (dashed, green); GOODE standard deviation (dotted, green). The latter is amplified with 10^3 for visualization purpose.

Note that incorporation of numerical approximation error via PN in inverse problems is still an active field of research and up to date no off-the-shelf methods are available. Recent work by Kersting et al. [Kersting et al., 2020] proposes a new method for ODE inverse problems. There Gaussian ODE filtering is applied to construct a local Gaussian approximation to the likelihood, which reflects epistemic uncertainty from numerical approximation. With the assumption that the ODE function is linear in its parameters, gradients and Hessian of the likelihood are then tractable and allow existing gradient-based optimization and sampling methods.

Considering numerical approximation uncertainty in the inverse problem is impor-

tant for coarse approximations, but might be negligible in contrast to other uncertainty sources, if a fine discretization is possible. This is illustrated in [Kersting et al., 2020, Figure 2], where numerical approximation uncertainty aware and unaware likelihoods are compared for a coarse and fine time discretization of an ODE system (Lotka-Volterra).

In the following work we use a fine discretization, yielding a vanishing numerical error and thus, for sake of simplicity, do not explicitly account for numerical approximation error. In other cases one might consider the presented probabilistic numerical methods.

4.3. Bayesian inference with model discrepancy

As already reasoned in previous sections, the Bayesian approach to inverse problems, introduced in Section 3.1, is advantageous to address the inverse problem and consider all sources of uncertainty. In (3.1) $\boldsymbol{\varepsilon}$ models both the measurement and model error. However it is often simply modeled as an identically and independent distributed (i.i.d.) Gaussian. I.e. for unknown standard deviations $\sigma_I, \sigma_\omega > 0$, the identity matrix $\boldsymbol{I}_{N_t} \in \mathbb{R}^{N_t \times N_t}$ and the covariance matrix $\Sigma(\sigma_I, \sigma_\omega) = \operatorname{diag}(\sigma_I^2 \boldsymbol{I}_{N_t}, \sigma_\omega^2 \boldsymbol{I}_{N_t}) \in \mathbb{R}^{2N_t \times 2N_t}$ let

$$\boldsymbol{\varepsilon}(\sigma_I, \sigma_\omega) = [\boldsymbol{\varepsilon}_I(\sigma_I), \boldsymbol{\varepsilon}_\omega(\sigma_\omega)]^T \sim \mathcal{N}(\mathbf{0}, \Sigma(\sigma_I, \sigma_\omega)). \tag{4.4}$$

The simplest model, denoted by **Bayesian model 1 (BM1)**, is then

$$\boldsymbol{y} = \mathcal{M}(\boldsymbol{x}) + \boldsymbol{\varepsilon}(\sigma_I, \sigma_\omega). \tag{4.5}$$

The likelihood, i.e. the distribution of the measurements $\boldsymbol{y}$ conditioned on the parameters $\boldsymbol{x}, \sigma_I$ and σ_ω, is

$$\boldsymbol{y}|\boldsymbol{x}, \sigma_I, \sigma_\omega \sim \mathcal{N}(\mathcal{M}(\boldsymbol{x}), \Sigma(\sigma_I, \sigma_\omega)). \tag{4.6}$$

With a prior distribution $\pi(\boldsymbol{x}, \sigma_I, \sigma_\omega) = \pi(\boldsymbol{x})\pi(\sigma_I)\pi(\sigma_\omega)$, expressing a-priori knowledge of the unknown parameters, Bayes' formula yields for the posterior distribution

$$\pi(\boldsymbol{x}, \sigma_I, \sigma_\omega|\boldsymbol{y}) \sim \pi(\boldsymbol{y}|\boldsymbol{x}, \sigma_I, \sigma_\omega)\pi(\boldsymbol{x}, \sigma_I, \sigma_\omega). \tag{4.7}$$

BM1 with the i.i.d. Gaussian measurement noise in principle assumes that the model is perfect and thus neglects model discrepancy. In contrast to this, we now follow the KO approach [Kennedy and O'Hagan, 2001] by explicitly considering and modeling the unknown discrepancy $\boldsymbol{\delta}$ between data $\boldsymbol{y}$ and simulation model $\mathcal{M}(\boldsymbol{x}^\dagger)$, where $\boldsymbol{x}^\dagger \in \mathcal{X}$ are optimal but unknown parameters. The resulting task is to simultaneously find model parameters $\boldsymbol{x} \in \mathcal{X}$, model discrepancy $\boldsymbol{\delta} = [\boldsymbol{\delta}_I, \boldsymbol{\delta}_\omega] \in \mathbb{R}^{2\times N_t}$ and measurement noise $\boldsymbol{\varepsilon}(\sigma_I, \sigma_\omega) \in \mathbb{R}^{2\times N_t}$ such that

$$\boldsymbol{y} = \mathcal{M}(\boldsymbol{x}) + \boldsymbol{\delta} + \boldsymbol{\varepsilon}(\sigma_I, \sigma_\omega). \tag{4.8}$$

For $D = [t_1, t_{N_t}]$ and unknown functions $\delta_I, \delta_\omega : D \to \mathbb{R}$ the model discrepancy vectors are

$$\boldsymbol{\delta}_I = [\delta_I(t_1), \dots, \delta_I(t_{N_t})], \quad \boldsymbol{\delta}_\omega = [\delta_\omega(t_1), \dots, \delta_\omega(t_{N_t})]. \tag{4.9}$$

In principle the unknown model discrepancy $\boldsymbol{\delta}$ can easily be included within the Bayesian framework, by defining a prior distribution and inferring it as well. However, it poses further challenges which will be detailed in the following section. Inference of each component in $\boldsymbol{\delta} \in \mathbb{R}^{2\times N_t}$ is in principle possible, but is not advisable as it does not scale. Consequently, additionally to the parametric form of the measurement noise term $\boldsymbol{\varepsilon}$, a parameterization of the model discrepancy functions δ_I, δ_ω is required. We introduce in the following Gaussian Process and orthogonal function approximations for the model discrepancy, due to the low number of parameters, but emphasize that for other applications other universal approximators (e.g. see Table 3.1) might be more suitable.

Remark. Additive measurement noise and model discrepancy is often a standard assumption. Of course, one should carefully examine for the considered problem, if this is justified. Otherwise one might need to transform the problem or use another noise assumption tailored to the problem and the available information about the error structure. In principle, a multiplicative model discrepancy term $\boldsymbol{y} = \mathcal{M}(\boldsymbol{x}) * \boldsymbol{\delta} + \boldsymbol{\varepsilon}$, where $*$ denotes element wise multiplication, is also possible, see e.g. [Edeling et al., 2014b, Edeling et al., 2014a]. However, the likelihood that arises from a multiplicative error term is more complicated [Dunlop, 2019]. For the problems considered in this work an additive error structure is sufficient. Further the Gaussian i.i.d. noise assumption needs to be justified. For instance, if there is evidence, it could be modeled as temporal or output correlated or with time dependent standard deviations $\sigma_I(t), \sigma_\omega(t) > 0$. Also one might think about a Laplace distribution that has fatter tails and is more forgiving for extreme events.

4.3.1. Specific challenges: identifiability problem and model selection

Going away from the trivial choice of only i.i.d. noise allows a more detailed quantification of uncertainties, but at the same time introduces additional challenges to the already existing ones of solving an inverse problem. The primary problem of identification requires modeling choices to be made for the discrepancy and consequently the difficult task of model selection. This on the other hand implies an increase of computational costs.

Identifiability problem

Primarily, there is the identifiability problem defined in 9. In contrast to the very rigid statistical model BM1, the approach with model discrepancy is very flexible, which

makes it impossible to identify a best parameter $\boldsymbol{x}^\dagger$ and a best model discrepancy $\boldsymbol{\delta}^\dagger$ without further information. In [Brynjarsdóttir and O'Hagan, 2014] they state that in order to infer model discrepancy and model parameters at the same time, at least for one of those an informative prior must be given. However, it is not clear how specific the prior information need to be to obtain an identifiable problem. Also, it implies that the inference result is way more sensitive on the prior, compared to BM1, where the prior has low influence for informative data $\boldsymbol{y}$ [Tuo and Wu, 2015, Tuo and Wu, 2016].

Model selection

A first step towards specific prior information is to introduce a parametric model for the model discrepancy term $\boldsymbol{\delta}$ with, at best, a low number of variables for inference. Let $\boldsymbol{\delta}(\boldsymbol{\theta})$ denote such a parametric model for $\boldsymbol{\delta}$ with parameters $\boldsymbol{\theta} \in \mathbb{R}^p$ and prior distribution $\pi(\boldsymbol{\theta})$. For competing variants of $\boldsymbol{\delta}(\boldsymbol{\theta})$ model selection is required. In principle model selection can also be done on the level of the simulation model, but assume for the following a fixed simulation model $\mathcal{M}$ and consider the selection of the statistical model only. Important factors that need to be balanced in order to specify and select the model discrepancy term are:

- **Identifiability**, i.e. $\boldsymbol{\delta}(\boldsymbol{\theta})$ and $\pi(\boldsymbol{\theta})$ should contain enough specific information to make inference of $\boldsymbol{x}, \boldsymbol{\theta}$ and σ_I, σ_ω identifiable;
- **Accuracy**, i.e. the parametric model $\boldsymbol{\delta}(\boldsymbol{\theta})$ should be able to approximate the true underlying discrepancy $\boldsymbol{\delta}$ as good as possible, which requires a certain generality or flexibility of $\boldsymbol{\delta}(\boldsymbol{\theta})$. However the generality needs to be balanced with the restrictions for identifiability;
- **Bias-variance trade-off** is the fact that with increasing model complexity of $\boldsymbol{\delta}(\boldsymbol{\theta})$ (but in principle also of $\mathcal{M}$) the bias $= \mathbb{E}[\boldsymbol{x}|\boldsymbol{y}] - \boldsymbol{x}^\dagger$ (difference between the mean of the estimator and the reference value) decreases, but at the same time the variance of the estimator $V[\boldsymbol{x}|\boldsymbol{y}]$ increases with the model complexity [Hastie et al., 2009]. Consequently, the mean square error $\mathrm{MSE}(\boldsymbol{x}) = \mathrm{bias}^2 + V[\boldsymbol{x}|\boldsymbol{y}]$ is minimal for an optimal model complexity, illustrated in Figure 4.2. With this optimal model complexity the contributions of bias and variance are somehow balanced. Models with a complexity over this optimum overfit (explain the measurement data too good) and below underfit (explain the measurement data too bad).
- **Computational costs**, i.e. the evaluation of $\boldsymbol{\delta}(\boldsymbol{\theta})$ should be with low computational costs and the number of additional parameters $\boldsymbol{\theta}$ should be low to reduce sampling effort.

[1] Figure source: http://scott.fortmann-roe.com/docs/BiasVariance.html

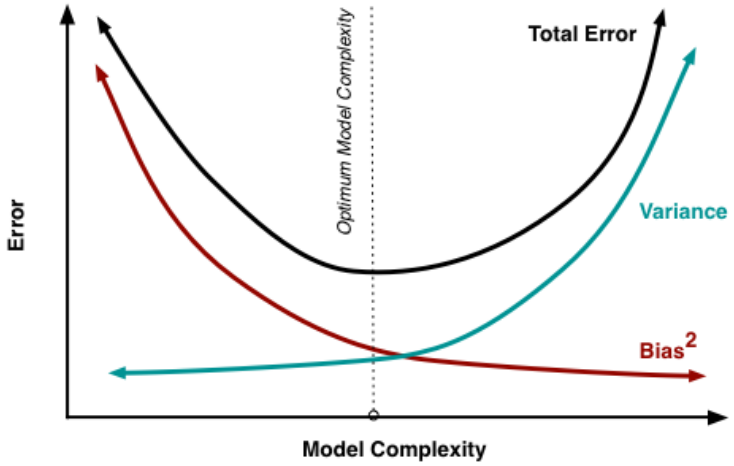


Figure 4.2.: Visualization of the bias-variance trade-off[1]. A dilema for supervised learning algorithms is to simultaneously minimize two error sources, the bias and the variance, which prevent generalization beyond the trainig data. Bias error occurs from wrong or unsufficient assumptions in the algorithm. Variance error is due to sensitivity to noise in the training data. With increasing model complexity the bias decreases, but at the same time the variance increases, and vice versa [Hastie et al., 2009].

4.3.2. Gaussian Process approximation

Kennedy and O'Hagan [Kennedy and O'Hagan, 2001] suggest to model the model discrepancy by a Gaussian Process

$$\delta(t) \sim \mathcal{GP}(m(t), k(t, t')), \tag{4.10}$$

with mean function $m : D \to \mathbb{R}$ and covariance function $k : D \times D \to \mathbb{R}$ (see Section 3.3 for an introduction to GPs). This kind of modeling is widely adopted in literature, as already presented in Section 4.1, e.g. in [Brynjarsdóttir and O'Hagan, 2014, Plumlee, 2017].

Assume $m(t) \equiv 0$ and a parametric covariance kernel $k_{(\cdot)}(t, t'; \boldsymbol{\theta}_{(\cdot)})$, depending on hyper-parameters $\boldsymbol{\theta}_{(\cdot)} \in \mathbb{R}^p$, respectively for I and ω. The GPs $\delta_{(\cdot)}(t)$ evaluated at time points $\Delta = \{t_1, \ldots, t_{N_t}\}$ are multivariate Gaussian and denoted by

$$\boldsymbol{\delta}_{(\cdot)}(\boldsymbol{\theta}_{(\cdot)}) \sim \mathcal{N}(\mathbf{0}, K^{(\cdot)}_{\Delta,\Delta}(\boldsymbol{\theta}_{(\cdot)})), \tag{4.11}$$

with covariance matrix elements $[K^{(\cdot)}_{\Delta,\Delta}(\boldsymbol{\theta}_{(\cdot)})]_{i,j} = k_{(\cdot)}(t_i, t_j; \boldsymbol{\theta}_{(\cdot)})$ for $t_i, t_j \in \Delta$. Then

$$\boldsymbol{\delta}(\boldsymbol{\theta}) = [\boldsymbol{\delta}_I(\boldsymbol{\theta}_I), \boldsymbol{\delta}_\omega(\boldsymbol{\theta}_\omega)]^T \in \mathbb{R}^{2\times N_t} \tag{4.12}$$

denotes the approximation of the true underlying model discrepancy $\boldsymbol{\delta}$ in (4.8), depending on both covariance functions k_I, k_ω and the hyper-parameters $\boldsymbol{\theta} = [\boldsymbol{\theta}_I, \boldsymbol{\theta}_\omega]$.

Now, with the Gaussian noise model for $\boldsymbol{\varepsilon}$ as in BM1, the unknown latent $\boldsymbol{\delta}$ can be analytically integrated out from the product of the GP prior distribution $\pi(\boldsymbol{\delta}|\boldsymbol{\theta})$ and the Gaussian likelihood $\pi(\boldsymbol{y}|\boldsymbol{x}, \sigma_I, \sigma_\omega, \boldsymbol{\delta}, \boldsymbol{\theta})$. The marginal likelihood

$$\pi(\boldsymbol{y}|\boldsymbol{x}, \sigma_I, \sigma_\omega, \boldsymbol{\theta}) = \int \pi(\boldsymbol{y}|\boldsymbol{x}, \sigma_I, \sigma_\omega, \boldsymbol{\delta}, \boldsymbol{\theta})\pi(\boldsymbol{\delta}|\boldsymbol{\theta})d\boldsymbol{\delta} \tag{4.13}$$

is then a multivariate Gaussian

$$\boldsymbol{y}|\boldsymbol{x}, \sigma_I, \sigma_\omega, \boldsymbol{\theta} \sim \mathcal{N}(\mathcal{M}(\boldsymbol{x}), K_{\Delta,\Delta}(\boldsymbol{\theta}) + \Sigma(\sigma_I, \sigma_\omega)). \tag{4.14}$$

And for given prior distribution $\pi(\boldsymbol{x}, \sigma_I, \sigma_\omega, \boldsymbol{\theta})$ the posterior distribution is given by

$$\pi(\boldsymbol{x}, \sigma_I, \sigma_\omega, \boldsymbol{\theta}|\boldsymbol{y}) \sim \pi(\boldsymbol{y}|\boldsymbol{x}, \sigma_I, \sigma_\omega, \boldsymbol{\theta})\pi(\boldsymbol{x}, \sigma_I, \sigma_\omega, \boldsymbol{\theta}). \tag{4.15}$$

The model discrepancy posterior distribution is $\delta_{(\cdot)}(t) \sim \mathcal{GP}(\underline{m}_{(\cdot)}(t), \underline{k}_{(\cdot)}(t, t'))$ with

$$\underline{m}_{(\cdot)}(t) = K^{(\cdot)}_{t,\Delta}(K^{(\cdot)}_{\Delta,\Delta})^{-1}\boldsymbol{y}^{(\cdot)}, \tag{4.16}$$

$$\underline{k}_{(\cdot)}(t, t') = K^{(\cdot)}_{t,t'} - K^{(\cdot)}_{t,\Delta}(K^{(\cdot)}_{\Delta,\Delta})^{-1}\boldsymbol{y}^{(\cdot)}K^{(\cdot)}_{\Delta,t'}, \tag{4.17}$$

where the dependency on $\boldsymbol{\theta}_{(\cdot)}$ is omitted for convenience.

Usually the log-marginal likelihood

$$\log \pi(\boldsymbol{y}|\boldsymbol{x}, \sigma_I, \sigma_\omega, \boldsymbol{\theta}) = -\frac{1}{2}(\boldsymbol{y} - \mathcal{M}(\boldsymbol{x}))^T \left(K_{\Delta,\Delta}(\boldsymbol{\theta}) + \Sigma(\sigma_I, \sigma_\omega)\right)^{-1} (\boldsymbol{y} - \mathcal{M}(\boldsymbol{x})) \quad (4.18)$$

$$- \frac{1}{2}\log |K_{\Delta,\Delta}(\boldsymbol{\theta}) + \Sigma(\sigma_I, \sigma_\omega)| - \frac{N_t}{2}\log(2\pi) \quad (4.19)$$

is used for inference. Here the first term stands for the data fit and the second term with the determinate for model complexity [Rasmussen and Williams, 2006]. Optimizing $\boldsymbol{\theta}$ w.r.t. the log-marginal likelihood automatically enforces model parsimony and avoids overfitting for the given parametric covariance kernel and mean function.

Evaluation of the marginal likelihood requires inversion of the covariance matrix, which is usually done via Cholesky decomposition. During sampling this is required for each sample of $\boldsymbol{\theta}$, which fast becomes prohibitively expensive for large N_t, as the cost for training scales with $\mathcal{O}(N_t^3)$. An approach to reduce the computational costs is to decrease the number of time points for inference and finally compute the conditional distribution to obtain a result for all time points. The reduced set of points are called inducing points Δ^{ind} and are either fixed or simultaneously optimized, see e.g. [Quiñonero-Candela and Rasmussen, 2005, Titsias, 2009] and the references therein for further details on sparse GP regression. With this the cost for training reduces to $\mathcal{O}(N_t n_t^2)$, where $n_t = |\Delta^{ind}|$ is the number of inducing points with $n_t < N_t$. For simplicity we use a fixed set of equidistant inducing points $\Delta^{ind} \subset \Delta$ and the Fully Independent Training Conditional (FITC) method for approximation in the following.

Assuming smoothness for δ we select a Matern 3/2 covariance kernel

$$k(t, t') = \tau^2 \left(1 + \frac{\sqrt{3}|t - t'|}{l}\right) \exp\left(-\frac{\sqrt{3}|t - t'|}{l}\right), \quad (4.20)$$

with standard deviation $\tau > 0$ and characteristic length scale $l > 0$. Figure 4.3 presents in the bottom row an overview of the distribution of a GP with such a kernel for $\tau = 1$ and $l = 4, 2, 1$.

For the following numerical experiments we denote

$$\boldsymbol{y} = \mathcal{M}(\boldsymbol{x}) + \boldsymbol{\delta}^{\mathcal{GP}}(\boldsymbol{\theta}) + \boldsymbol{\varepsilon}(\sigma_I, \sigma_\omega) \quad (4.21)$$

as **Bayesian model with GPs (GP)** with likelihood as above and model discrepancy parameters $\boldsymbol{\theta} = [\tau_I, \tau_\omega, l_I, l_\omega]$.

4.3.3. Orthogonal function approximation

Now, the model discrepancy is modeled by a truncated expansion of orthogonal functions, as in [Nagel et al., 2017]. However, we simplify the correlated noise assumption made in [Nagel et al., 2017] to an i.i.d. noise assumption for $\boldsymbol{\varepsilon}$ for the benefit of computational

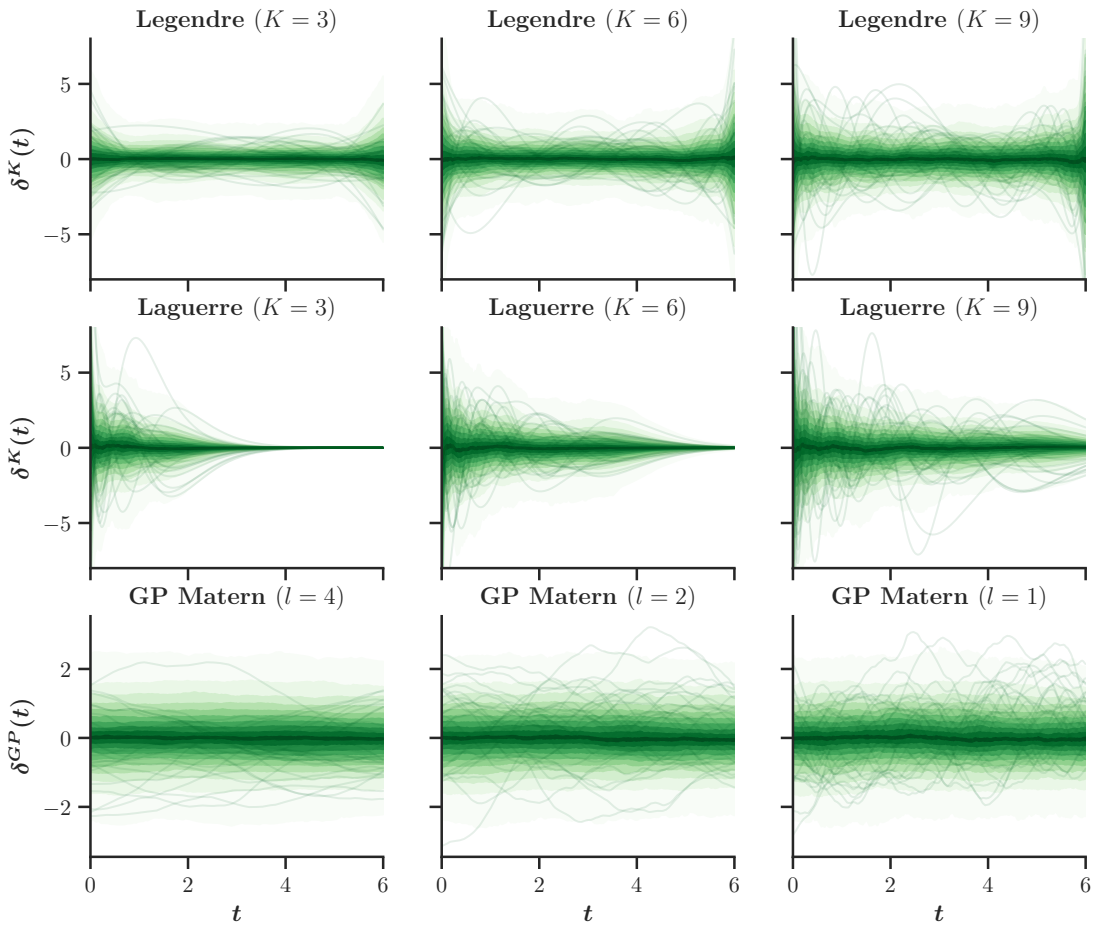


Figure 4.3.: Prior distribution for model discrepancy $\boldsymbol{\delta}$ modeled with orthogonal functions $\boldsymbol{\delta}^K$ (top row: Legendre polynomials on $[0, 6]$, middle row: weighted Laguerre polynomials with scale $s = 6$) with maximal polynomial degree K and Gaussian Process $\boldsymbol{\delta}^{\mathcal{GP}}$ with Matern 3/2 kernel and length scale l.

efficiency. Yielding a cost efficient alternative for model discrepancy approximation in comparison to the cost intensive GP approximation. Further, we suggest an iterative approach to select the truncation order that is intended to infer physical meaningful parameters and weaken identifiability problems.

For orthogonal function approximation we assume $\delta \in L^2(D)$ and model it by a truncated expansion of functions which are orthogonal with respect to the inner product $\langle f, g \rangle = \int_D fg d\mu$, where μ is a measure on D and $f, g \in L^2(D)$. Let $\{p_j\}_{j\in\mathbb{N}} \subseteq L^2(D)$ be a basis of functions $p_j : D \to \mathbb{R}$ dense in $L^2(D)$, i.e. $span(\{p_j\}_{j\in\mathbb{N}}) = L^2(D)$. Then for all $\delta \in L^2(D)$ there exists $\{a_j\}_{j\in\mathbb{N}} \subseteq \mathbb{R}$ with $\sum_{j\in\mathbb{N}} |a_j|^2 < \infty$ such that δ can be represented by the expansion

$$\delta(t) = \sum_{j=0}^{\infty} a_j p_j(t). \tag{4.22}$$

For practicability reasons the expansion is truncated after a $K \in \mathbb{N}$

$$\delta(t) \approx \delta^K(t) = \sum_{j=0}^{K} a_j p_j(t). \tag{4.23}$$

Let $\delta_I^K(t)$ and $\delta_\omega^K(t)$ denote the approximative models for the model discrepancy terms δ_I, δ_ω, with coefficients $a_j^{(I)}$ and $a_j^{(\omega)}$ for $j = 0, \dots, K$, respectively. Note that the basis $\{p_j\}_{j=0,\dots,K}$ and truncation parameter K do not necessarily need to be identical. Let

$$\boldsymbol{a} = [\boldsymbol{a}^{(I)}, \boldsymbol{a}^{(\omega)}] = [a_0^{(I)}, \dots, a_K^{(I)}, a_0^{(\omega)}, \dots, a_K^{(\omega)}] \in \mathbb{R}^{2K+2} \tag{4.24}$$

be the vector containing all coefficients and $\boldsymbol{\delta}_I^K(\boldsymbol{a}) = [\delta_I^K(t_0), \dots, \delta_I^K(t_{N_t})]$, $\boldsymbol{\delta}_\omega^K(\boldsymbol{a}) = [\delta_\omega^K(t_1), \dots, \delta_\omega^K(t_{N_t})]$ be the vectors of the model discrepancy terms evaluated at the discrete time points $t_i, i = 1, \dots, N_t$. Hence

$$\boldsymbol{\delta}^K(\boldsymbol{a}) = [\boldsymbol{\delta}_I^K(\boldsymbol{a}), \boldsymbol{\delta}_\omega^K(\boldsymbol{a})]^T \in \mathbb{R}^{2\times N_t} \tag{4.25}$$

denotes the approximation of the true underlying model discrepancy $\boldsymbol{\delta}$. The **Bayesian model 2 (BM2)** with all parameter dependencies is

$$\boldsymbol{y} = \mathcal{M}(\boldsymbol{x}) + \boldsymbol{\delta}^K(\boldsymbol{a}) + \boldsymbol{\varepsilon}(\sigma_I, \sigma_\omega). \tag{4.26}$$

The number of unknown parameters depends on K and is $n + 2K + 2 + 2$. With the additional unknown coefficients $\boldsymbol{a}$, the prior distribution is defined as

$$\pi(\boldsymbol{x}, \sigma_I, \sigma_\omega, \boldsymbol{a}) = \pi(\boldsymbol{x})\pi(\sigma_I)\pi(\sigma_\omega)\pi(\boldsymbol{a}), \tag{4.27}$$

where

$$\pi(\boldsymbol{a}) = \prod_{j=0}^{K} \pi(a_j^{(I)})\pi(a_j^{(\omega)}). \tag{4.28}$$

Now, with the likelihood

$$\boldsymbol{y}|\boldsymbol{x}, \sigma_I, \sigma_\omega, \boldsymbol{a} \sim \mathcal{N}(\mathcal{M}(\boldsymbol{x}) + \boldsymbol{\delta}^K(\boldsymbol{a}), \Sigma(\sigma_I, \sigma_\omega)), \tag{4.29}$$

the posterior is given by

$$\pi^K(\boldsymbol{x}, \sigma_I, \sigma_\omega, \boldsymbol{a}|\boldsymbol{y}) \sim \pi(\boldsymbol{y}|\boldsymbol{x}, \sigma_I, \sigma_\omega, \boldsymbol{a})\pi(\boldsymbol{x}, \sigma_I, \sigma_\omega, \boldsymbol{a}), \tag{4.30}$$

where π^K denotes the dependence on K.

Remark. In contrast to the previous section, where $\delta(t) \sim \mathcal{GP}(m(t), k(t, t'))$ with $m(t) \equiv 0$ and a parametric model for $k(t, t')$, here the mean function is explicitly modeled by the truncated function expansion and, in principle, $k(t, t') \equiv 0$. The temporal dependence of δ is now modeled by a weighted combination of the orthogonal basis functions temporal dependence. The covariance matrix simplifies to a diagonal matrix based on the i.i.d. measurement noise $\boldsymbol{\varepsilon}$, resulting in low computational costs for likelihood evaluation. Of course both approaches could be combined by modeling both, the mean function and the covariance, but this again would require inducing points to reduce computational costs for inversion.

For further proceeding the basis functions $\{p_j\}_{j=0,\dots,K}$, the prior for the coefficients $\pi(\boldsymbol{a})$ and the truncation parameter K need to be specified. If knowledge about the discrepancy is available, this should be modeled accordingly by defining an appropriate prior distribution for $\boldsymbol{\delta}^K(\boldsymbol{a})$. However, in general this knowledge is not available and some modeling assumptions need to be made. For these the model discrepancy related specific challenges detailed in Section 4.3.1 need to be considered in order to balance identifiability, accuracy (model complexity), bias-variance trade-off and computational costs.

Basis functions

Following the assumption above, the basis needs to be dense in $L^2(D)$. With the additional assumption that $\boldsymbol{\delta}$ is rather smooth, polynomials are a reasonable choice. For this let $p_j : D \to \mathbb{R}$ be a polynomial with polynomial degree $deg(p_j) = j$ and $\{p_j\}_{j\in\mathbb{N}}$ be an orthonormal polynomial basis. Let $w : D \to \mathbb{R}_+$ be the weighting function of μ. Then, for instance, the

- (a) Legendre polynomials for $w(t) = \dfrac{1}{|D|}$, with $|D| = t_{N_t} - t_1$

- (b) Tschebyschew polynomials for $w(t) = \frac{1}{\sqrt{1-t^2}}$ and
- (c) Laguerre polynomials for $w(t) = \exp^{-t}$

are, among others, dense in $L^2(D)$ and thus possible choices for the expansion. Legendre polynomials with the constant weighting function are a reasonable choice for $\{p_j\}_{j=0,\dots,K}$. This choice was also made in [Nagel et al., 2017].

Another option are the scaled and weighted Laguerre polynomials $\{p_j\}_{j\in\mathbb{N}}$, weighted with the square root of their respective weighting function, i.e. $\phi_j(st) = p_j(st)\exp(-st/2)$ for scaling $s > 0$. For s fixed and $t \to \infty$ the $\phi_j(st) \to 0$, which consequently implies the same for an expansion in $\{\phi_j\}_{j\in\mathbb{N}}$ with fixed expansion coefficients. This encodes that the model discrepancy tends to zero for large t.

Figure 4.3 displays an overview of the distribution of $\boldsymbol{\delta}^K$ modeled with (top row) Legendre polynomials on $[0, 6]$ and (middle row) weighted Laguerre polynomials with scale $s = 6$ for $K = 3, 6, 9$ (columns) and coefficients sampled as detailed below. Note that the variance in both cases is non-constant and varies with t, i.e. $V[\delta^K_{(\cdot)}(t)] = \sum_{j=0}^{K} V[a_j^{(\cdot)}]p_j^2(t)$.

Prior distributions for the coefficients

The choice of the prior distribution for the coefficients is a bit delicate and somewhat arbitrary, as they have no physical meaning. Since no knowledge about the model discrepancy $\boldsymbol{\delta}$ is available we assume $\boldsymbol{\delta}^K(\boldsymbol{a})$ to be as small as possible by specifying priors that are centered around zero. Every around zero centered probability distribution with decaying tails should be sufficient. However, following [Nagel et al., 2017], we opt for zero mean Laplace distributions, as they assign the highest probability around zero and decay exponentially towards the tails. The probability density function of $Laplace(a, b)$ is $f(x) = \frac{1}{2b}exp(-\frac{|x-a|}{b})$ for $x \in \mathbb{R}$ [Kotz et al., 2012]. Compared to a Gaussian distribution, Laplace distribution enforces the zero a bit more and has slightly fatter tails, see Figure 4.4 for a visual comparison. Consequently, the $\pi(a_j^{(I)}), \pi(a_j^{(\omega)})$ are modeled as Laplace distributions with zero mean and variance $b > 0$. As the coefficients $a_j^{(I)}, a_j^{(\omega)}$ have independent zero mean priors the resulting distributions for $\boldsymbol{\delta}_I^K, \boldsymbol{\delta}_\omega^K$ are centered around zero as well, visualized in Figure 4.3 where $b = 10$.

Truncation parameter K

Assume that a polynomial basis and the prior for the coefficients are given, then an appropriate truncation parameter K needs to be selected. With the choice of orthogonal polynomials, K corresponds to the maximum polynomial degree of the basis polynomials and determines the complexity of the model discrepancy term $\boldsymbol{\delta}^K(\boldsymbol{a})$. With increasing K the discrepancy term $\boldsymbol{\delta}^K(\boldsymbol{a})$ yields an increased flexibility and is able to approximate a growing class of functions. However, a remedy of this increased flexibility is a loss

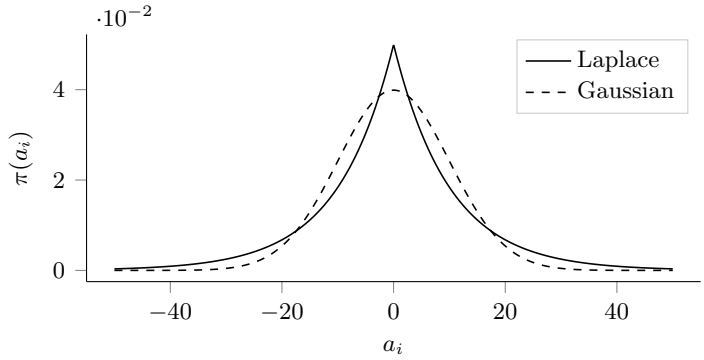


Figure 4.4.: The probability density function of a Laplace distribution $Laplace(0, 10)$ compared to a Gaussian distribution $\mathcal{N}(0, 10^2)$.

of information in the prior of $\boldsymbol{\delta}^K(\boldsymbol{a})$. This might yield identifiability problems for all unknown parameters, depending on the information content of the remaining model parameter prior distributions. Furthermore, a large K might yield overfitting, i.e. high degree polynomials inadvertently start to reproduce oscillations of the measurement noise. With respect to computational costs K should be as small as possible to reduce the number of unknown parameters to sample from. Consequently, an optimal K should be just large enough such that $\boldsymbol{\delta}^K(\boldsymbol{a})$ is accurate enough to approximate the underlying discrepancy correct, while reducing identifiability and overfitting problems. Finally, taking all these factors into account the suggested approach in this work on how to find an optimal K^{opt} is sketched in Algorithm 5: Start with an initial $K = 0$ and increase K

Algorithm 5: Selection of optimal truncation parameter K^{opt}

Initialize: $K = 0,\ tol > 0,\ \kappa^{max} \in \mathbb{N}$
1. Compute $\pi^K(\sigma_I, \sigma_\omega|\boldsymbol{y}),\ \ldots,\ \pi^{K+\kappa^{max}}(\sigma_I, \sigma_\omega|\boldsymbol{y})$
2. If $D(\pi^K(\sigma_I, \sigma_\omega|\boldsymbol{y}), \pi^{K+\kappa}(\sigma_I, \sigma_\omega|\boldsymbol{y})) < tol$ for $\kappa = 1, \ldots, \kappa^{max}$
2.1. Set $K^{opt} = K$ and stop.
3. Else
3.1. Set $K = K + 1$
3.2. Compute $\pi^{K+\kappa^{max}}(\sigma_I, \sigma_\omega|\boldsymbol{y})$
3.3. Return to 2.

iteratively until the marginal posterior distribution $\pi^K(\sigma_I, \sigma_\omega|\boldsymbol{y})$ of the noise standard deviation stabilizes, i.e. until the condition in line 2 holds for given distance measure $D(\cdot, \cdot)$, tolerance $tol > 0$ and maximum length κ^{max}.

Why is this sufficient? If a model discrepancy is present in BM1, then the noise term and the parameters are the only instances to capture it. As the noise term is modeled

with zero mean, the standard deviation might be overestimated consequently. By adding the model discrepancy term in BM2, it captures, depending on its flexibility and degrees of freedom, a fraction of the model discrepancy. As a consequence, the noise term $\boldsymbol{\varepsilon}$ needs to represent only the remaining discrepancy and is estimated by a smaller value. Due to the smoothness assumption of $\boldsymbol{\delta}^K(\boldsymbol{a})$, $\boldsymbol{\varepsilon}$ collects also everything that is not smooth enough. Now, if the estimated standard deviation of the noise does not change anymore from K to $K+\kappa$ for $\kappa = 1, \ldots, \kappa^{max}$ within the tolerance, then the smallest sufficient K is found. For this K the model discrepancy term $\boldsymbol{\delta}^K(\boldsymbol{a})$ should represent the underlying model discrepancy appropriately and a separation of model, parameter and measurement uncertainty is achieved.

Remark. In principle, K could be modeled by a discrete random variable in the Bayesian model and inferred as well. However this bears technical difficulties. E.g. for $K \in \{0, K_{max}\}$ there are active coefficients $a_j^{(\cdot)}, j = 0, \ldots, K$ and passive coefficients $a_j^{(\cdot)}, j = K+1, \ldots, K_{max}$. Active means that they are considered in $\boldsymbol{\delta}^K(\boldsymbol{a})$. If K switches during inference, then active coefficients get passive or vice versa. The problem is now how to handle this switching together with the often required burn in phase of sampling methods.

4.4. Sampling with surrogate-based gradient

Sampling from the posterior distribution is a difficult task, in particular for the complex statistical models of BM2 and GP, which include the complex physical model $\mathcal{M}$ in the likelihood. Sampling with MH-MCMC works, but is inefficient, due to high autocorrelation. For a decent approximation of the posterior distribution a large number of samples is required, implying long runtimes even with the usage of cheaper to evaluate surrogates for $\mathcal{M}$.

Often statistical models are expressed as symbolic computation graphs (e.g. with the Python library Theano [Theano Development Team, 2016] for the probabilistic programming library PyMC3 [Salvatier et al., 2016]), which generally allow automatic differentiation via efficient symbolic manipulation. Sampling efficiency can then be improved with advanced gradient-based samplers like Hamiltonian Monte Carlo (HMC) and No-U-turn sampler (NUTS) [Hoffman and Gelman, 2014]. However, complex physical models $\mathcal{M}$ or their surrogates are often black boxes, prohibiting full automatic differentiation.

One option are combined samplers, i.e. a gradient-based one for the automatic differentiable parameters combined with a gradient-free for all other parameters, e.g. for inference with BM2 the MH-MCMC sampler for $\boldsymbol{x}$ and NUTS for $\boldsymbol{a}, \sigma_I, \sigma_\omega$. This improves sampling, but there is still a bottleneck with the MH-MCMC step for $\boldsymbol{x}$.

In this work we use the fact that the derivatives of PCE surrogates $\mathcal{M}^{PCE}$ are analytically tractable and implement the Jacobian $\nabla\mathcal{M}^{PCE}(\boldsymbol{x})$ within the graph. In particular, $\mathcal{M}^{PCE}(\boldsymbol{x})$ is wrapped as Theano operator with a method that computes a

vector-Jacobian product, where $\nabla \mathcal{M}^{PCE}(\boldsymbol{x})$ itself is wrapped as a Theano operator (If an analytical gradient is not available, then finite differences are another option). This makes automatic differentiation possible and allows full usage of gradient-based samplers. The benefit is visualized in Figure 4.5, where samples from a combined sampler (MH-MCMC and NUTS) with high autocorrelation are compared to surrogate-based NUTS samples. Clearly the latter are superior w.r.t. autocorrelation, consequently requiring a lower number of total samples to approximate the posterior distribution properly. This is in line with the findings and illustration in [Hoffman and Gelman, 2014, Figure 7]. A drawback are the higher computational costs per sample for surrogate-based NUTS, as an additional evaluation of $\nabla \mathcal{M}^{PCE}(\boldsymbol{x})$ is required.

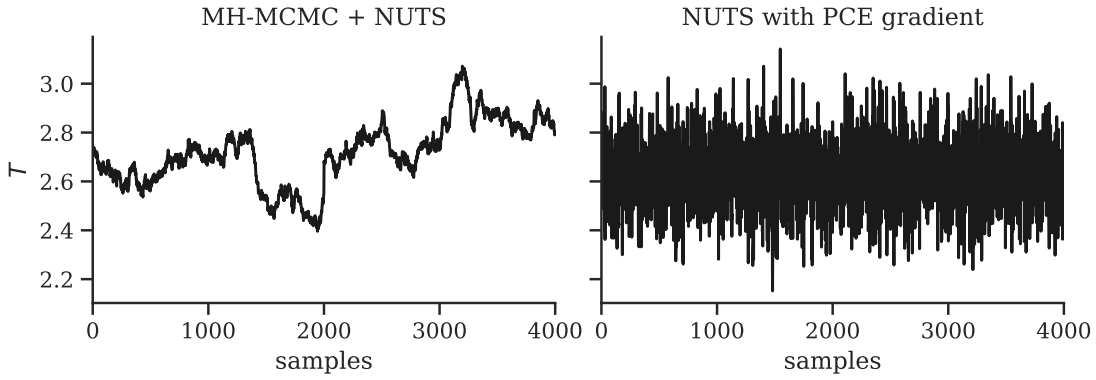


Figure 4.5.: Marginal posterior samples of parameter T obtained with combined MH-MCMC and NUTS sampling (left) and full NUTS sampling with surrogate-based gradient (right)[2].

Remark. Another (technical) option for implementation is to symbolically export $\mathcal{M}^{PCE}$ (e.g. with the Python library Sympy) and then to convert/import it as Theano graph, which, in principle, allows automatic differentiation. It works well for small PCE with only a few time points, but bears technical difficulties for a large PCE with many time points.

4.5. Numerical experiments

The numerical experiments are presented in following order: first for the basic electric motor model (see Section 2.1) with synthetic data and second for the test bench data with the corresponding basic model (2.11) (see Section 2.2). The applied methods are:

[2]This are already anticipated results from Section 4.5.2 for inference of V and T with BM2 (Legendre polynomials and $K = 3$) for synthetic data with discrepancy case linear.

- BM1 defined in (4.5), the simplest model with only i.i.d. Gaussian measurement noise $\boldsymbol{\varepsilon}(\sigma_I, \sigma_\omega)$;
- BM2 defined in (4.26), with orthogonal function model discrepancy $\boldsymbol{\delta}^K(\boldsymbol{a})$ additionally to BM1;
- GP defined in (4.21), with Gaussian Process model discrepancy $\boldsymbol{\delta}^{\mathcal{GP}}(\boldsymbol{\theta})$ additionally to BM1.

In the following Section 4.5.1 details on the joint inference setup. Then Section 4.5.2 and Section 4.5.3 present the numerical results for synthetic and test bench data, respectively. Section 4.5.4 concludes with a discussion of the results.

4.5.1. Inference setup

For all three methods the priors for the unknown standard deviations of the noise $\pi(\sigma_I), \pi(\sigma_\omega)$ are defined as uninformative Inverse Gamma distributions $InvGamma(\alpha = 2, \beta = 1)$. This is a common choice for conjugate priors of scale parameters in Bayesian statistics [Gelman et al., 2013], in particular for a Gaussian likelihood with given mean. The inverse Gamma distribution ensures the positiveness of $\sigma_I, \sigma_\omega > 0$.

The considered model parameters $\boldsymbol{x}$ are either the voltage V or the mechanical load T or both. The default prior distributions are $V \sim \mathcal{N}(13.5, 0.7^2)$ and $T \sim \mathcal{N}(2.5, 0.2^2)$. The default setting for BM2 are Legendre polynomials with $Laplace(0, b = 1)$ priors for the coefficients $\boldsymbol{a}$. Alterations of the default setting are noted accordingly in the following.

For GP the priors are for the length scales $l_I, l_\omega \sim Gamma(\alpha = 2, \beta = 1)$ and for the standard deviations $\tau_I, \tau_\omega \sim InvGamma(\alpha = 1, \beta = 1)$. The number of inducing points is $n_t = 31$ (in contrast to all time points with $N_t = 601$).

The simulation models are replaced by PCE surrogates for speed up, as detailed in Section 3.3.1. The PCE level L for the basic electric motor model is $L = 2$ and for the test bench model $L = 3$. For approximation of the posterior distributions the NUTS sampler with surrogate-based gradients is used, see Section 4.4. For BM1 and BM2 four parallel Markov Chains with 1500 samples each are sampled. For GP only 2 chains are sampled, due to higher computational effort. All computations are performed on a standard work station.

4.5.2. Synthetic data

Synthetic data allows the control of data generation and thus a detailed comparison and analysis of the inference results to available reference values. For synthetic data the basic electric motor model in Section 2.1 is used. The synthetic measurement data $\boldsymbol{y}$ is

generated by

$$\boldsymbol{y} = \mathcal{M}(\boldsymbol{x}^\dagger) + \boldsymbol{\delta}^\dagger + \boldsymbol{\varepsilon}^\dagger, \tag{4.31}$$

where the simulation model $\mathcal{M}$ is evaluated at reference model parameters $\boldsymbol{x}^\dagger$, distorted by a reference model discrepancy $\boldsymbol{\delta}^\dagger$ and corrupted by a sample of Gaussian measurement noise $\boldsymbol{\varepsilon}^\dagger$ with reference noise standard deviations $\sigma_I^\dagger = 0.1$ and $\sigma_\omega^\dagger = 0.5$, for outputs current I and angular velocity ω, respectively. $\boldsymbol{x}^\dagger$ are specified in Section 2.1 with $V^\dagger = 12$ and $T^\dagger = 2.5$.

As this chapter is all about model discrepancy, the following presents numerical experiments with varying definition of $\boldsymbol{\delta}^\dagger$ for the inference of model parameters V, T and model discrepancy $\boldsymbol{\delta}$.

Results for zero, constant, linear and quadratic $\delta^\dagger$

Table 4.1 lists four different model discrepancy cases with specific definitions of $\boldsymbol{\delta}^\dagger$. Figure 4.6 presents the marginal posterior distributions of V, σ_I and σ_ω (rows) for

	$\delta_I^\dagger(t)$	$\delta_\omega^\dagger(t)$
zero	0	0
constant	0.1	0.5
linear	$0.025t$	$0.1t$
quadratic	$t(t-5)/50$	$t(t-5)/100$

Table 4.1.: Reference model discrepancy $\boldsymbol{\delta}^\dagger = [\delta_I^\dagger(\Delta), \delta_\omega^\dagger(\Delta)]$ explicitly defined as zero, constant, linear or quadratic functions, respectively for current I and angular velocity ω.

discrepancy cases zero, constant, linear and quadratic (columns). Boxplots represent for each of these discrepancy cases the marginal posterior distributions obtained via GP, BM1 and BM2 for $K = 0, \ldots, 6$ (with Legendre polynomials and $b = 1$). The reference values are plotted for comparison. Note that the posterior distribution of σ_I, σ_ω does not necessarily need to correspond to $\sigma_I^\dagger, \sigma_\omega^\dagger$, but rather to empirical noise standard deviations (not visualized), due to the finite number of data points. This holds for all following figures. The BM1 solution for V is biased and overconfident for cases constant, linear and quadratic in comparison to the reference values. As expected σ_I and σ_ω are overestimated to compensate the discrepancy that cannot be handled by BM1. GP is in general more uncertain, but also biased for cases zero, constant and linear. Here, σ_I and σ_ω are well estimated due to GP discrepancy modeling. For BM2 the marginal posterior distributions of σ_I and σ_ω decrease in value for increasing K as the model discrepancy term takes over what previously was covered by the noise term only. BM2 with the

selection of an optimal K based on the marginal posterior distributions of σ_I, σ_ω and Algorithm 5 leads to better results for the posterior distribution of V. We also observe an increasing uncertainty and bias (note that the prior for V is centered at 13.5) in BM2 solutions for increasing, non-optimal K. The identified optimal K correspond to the polynomial degree of the reference discrepancy, namely $K = 0$ for cases zero and constant, $K = 1$ for case linear and $K = 2$ for case quadratic. The model discrepancy posterior distributions with those optimal K are displayed in Figure 4.7. They are less uncertain, compared to the ones obtained with GP. Similar results can be achieved for inference of T.

Inference of V and T simultaneously is more problematic as displayed in Figure 4.8. The results with respect to the posterior distributions of V and T are as expected for BM1: correct for case zero, but biased and overconfident for cases constant, linear and quadratic. Unfortunately, GP and BM2 results are biased as well and very uncertain. Here the posterior distributions are almost as uncertain as the prior distributions for V and T, but a bit shifted. Positive with BM2 is the correct identification of the optimal K based on Algorithm 5 and the marginal posterior distributions of σ_I and σ_ω. The optimal K's correspond to the polynomial degree of the reference discrepancies, at least identifying the shape of the model discrepancy correctly, but with an bias and high uncertainty. The BM2 and GP model discrepancy posterior distributions look almost similar to those in Figure 4.7, but with more uncertainty and bias.

Modifying the Laplace prior for the BM2 model discrepancy coefficients $\boldsymbol{a}$ from $b = 1$ to $b = 0.1$ has high influence on the posterior distributions for inference of V or T alone, but less influence for simultaneous inference of V and T. For V with $b = 0.1$ the results are displayed in Figure A.1 in the Appendix. The results of GP and BM1 are unaltered, but the BM2 results are now similar biased as BM1. Now, the uncertainty does not increase with K. Identification of optimal K is still possible, leading to model discrepancy posterior distributions that are correct in shape but slightly biased. Results for T alone are similar. For V and T with $b = 0.1$ the results are, in comparison to those with $b = 1$ in Figure 4.8, for BM2 generally only less uncertain and a bit less biased. Also here, identification of optimal K is still possible.

The joint inference of V and T together is problematic. From the sensitivity analysis of the basic electric motor model in Figure 3.4, it is known that both parameters have an influence on both outputs, only with varying share during time. Consequently, varying the input V, T varies the output I, ω indirectly w.r.t. the simulation model. Additionally, varying $\boldsymbol{\delta}_I, \boldsymbol{\delta}_\omega$ varies the output I, ω directly. Now, varying V, T and $\boldsymbol{\delta}_I, \boldsymbol{\delta}_\omega$ simultaneously leads to an identification problem. The effects of single variations are now not uniquely separable anymore, which makes the corresponding inverse problem extremely difficult to tackle. With this the prior distribution has increased influence on the posterior (as shown with reduction of b). Of course modifying the prior of V and T to be more informative would improve BM2 and also GP results correspondingly.

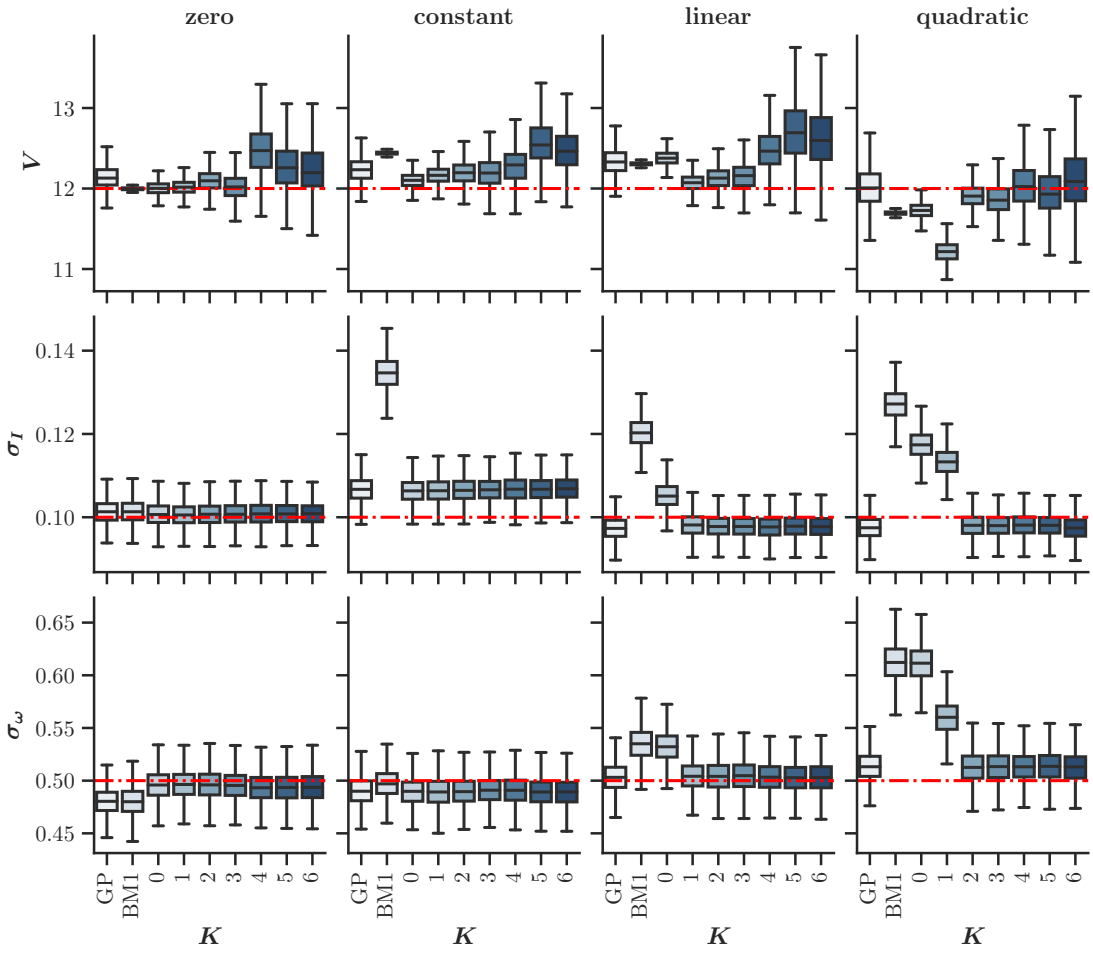


Figure 4.6.: Posterior distributions of V, σ_I and σ_ω (rows) for discrepancy cases zero, constant, linear and quadratic (columns). Results are obtained via GP, BM1 and BM2 for $K = 0, \ldots, 6$ (with $b = 1$). Reference values are red dash-dotted.

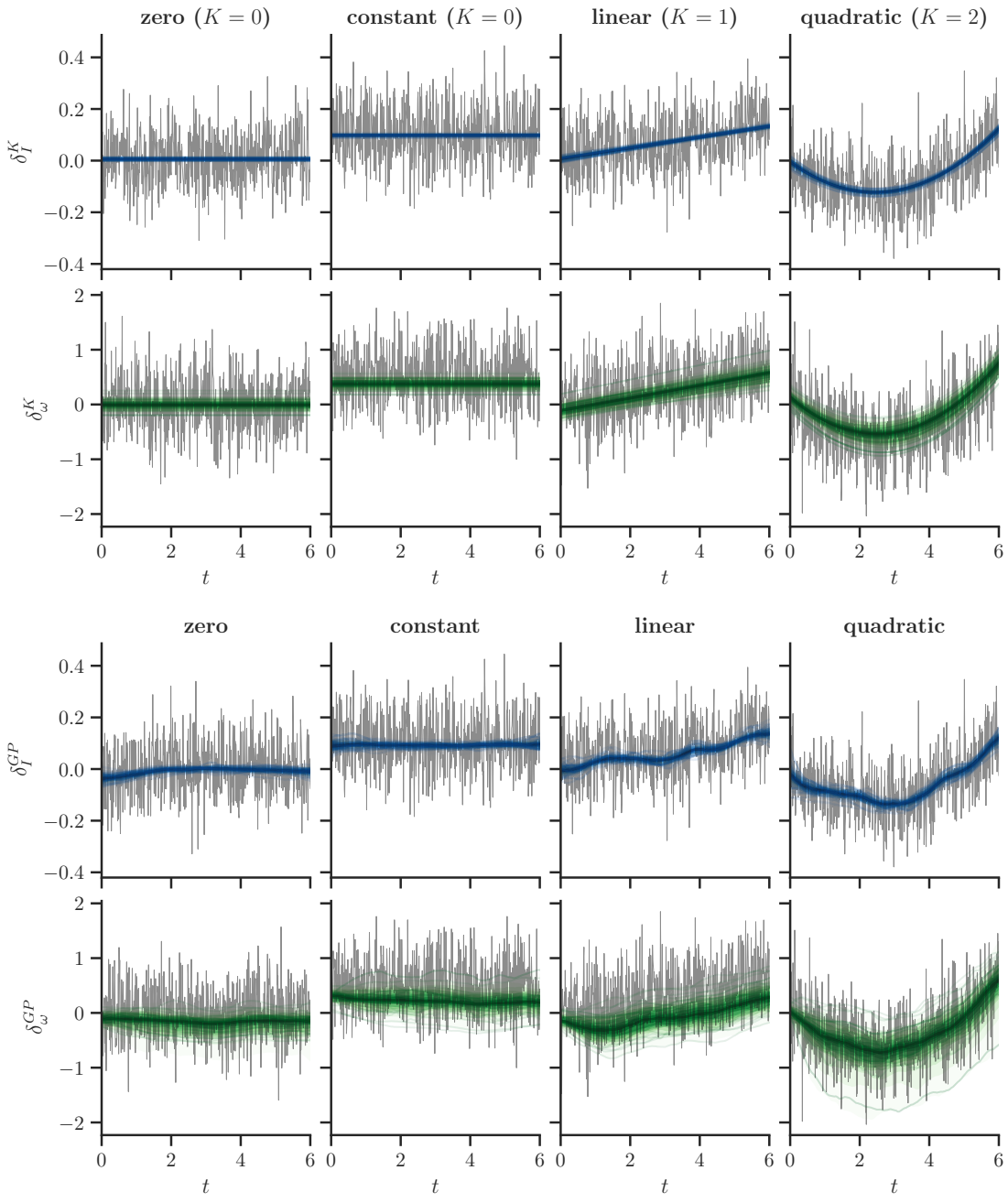


Figure 4.7.: Posterior distributions of δ_I, δ_ω (rows) for inference of V, σ_I and σ_ω (see Figure 4.6) for discrepancy cases zero, constant, linear and quadratic (columns). Results are obtained via BM2 with K specified in the heading (top half) and GP (bottom half). Reference discrepancy plus measurement noise is gray.

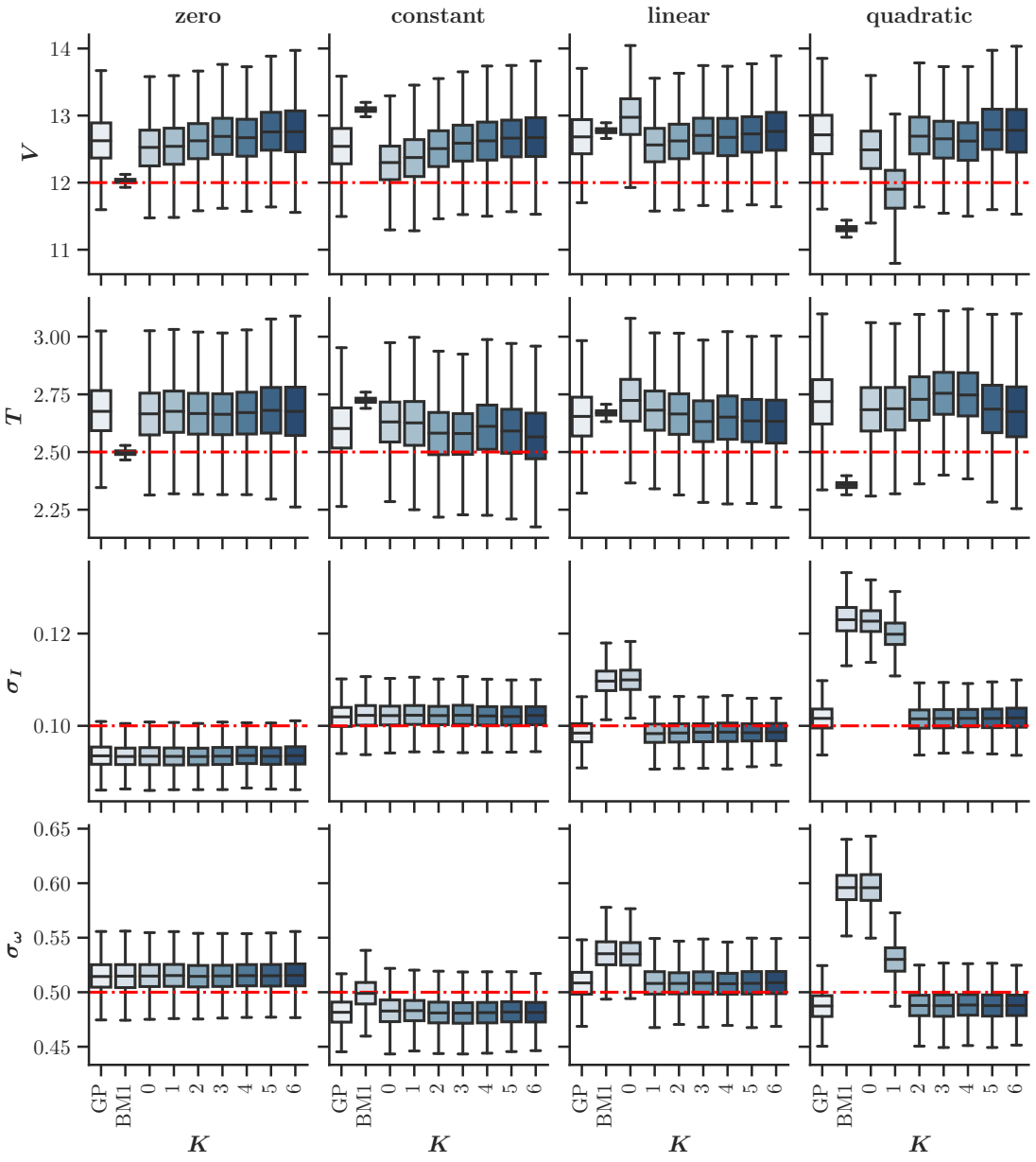


Figure 4.8.: Posterior distributions of V, T, σ_I and σ_ω (rows) for discrepancy cases zero, constant, linear and quadratic (columns). Results are obtained via GP, BM1 and BM2 for $K = 0, \ldots, 6$ (with $b = 1$). Reference values are red dash-dotted.

Remark. Anticipating the discussion later on, we already state an idea here for further research in order to improve this problem: a sensitivity informed likelihood. BM2 worked well for single parameters, but not for both. Thus splitting the likelihood in time w.r.t. time dependent sensitivity indices, for instance, might help to better account for single variations of the parameters. Research in this direction is unfortunately out of scope for this work.

Results for $\delta^\dagger$ based on model misspecification

Table 4.2 lists two different model discrepancy cases with implicit definitions of $\boldsymbol{\delta}^\dagger$ via misspecification of model parameters. This reference model discrepancy occurs by generating the synthetic data with unaltered reference parameters (right term in the sum) and performing inference with altered reference parameters (as in the left term of the sum).

	$[\boldsymbol{\delta}_I^\dagger, \boldsymbol{\delta}_\omega^\dagger]$
case c_g	$\mathcal{M}(x^\dagger; c_g = 0.9c_g^\dagger) - \mathcal{M}(x^\dagger; c_g = c_g^\dagger)$
case J	$\mathcal{M}(x^\dagger; J = 3J^\dagger) - \mathcal{M}(x^\dagger; J = J^\dagger)$

Table 4.2.: Reference model discrepancy $\boldsymbol{\delta}^\dagger = [\delta_I^\dagger, \delta_\omega^\dagger]$ implicitly defined via misspecification of model parameters.

The case c_g for inference of resistance R and model discrepancy $\boldsymbol{\delta}$ was already treated successfully in the author's previous publications, see [John et al., 2018a, John et al., 2018b, John et al., 2019b]. Due to this already detailed documentation, the results are not repeated here.

Based on the case J data the voltage V is inferred with methods GP, BM1 and BM2 for $K = 0, \dots, 12$. For BM2 three different specifications for the model discrepancy prior $\boldsymbol{\delta}^K(\boldsymbol{a})$ are used, i.e. with varying polynomials (Legendre and weighted Laguerre) and varying coefficient prior $\pi(\boldsymbol{a}) = Laplace(0, b)$. The posterior distributions for parameters V, σ_I, σ_ω are displayed in Figure 4.9 and for model discrepancy $\boldsymbol{\delta}$ in Figure 4.10.

The BM1 results for V are biased and overconfident. The GP results are even more biased and have high uncertainty. The GP model discrepancy posterior distribution properly infers the shape of the reference model discrepancy, but is biased and uncertain, see Figure 4.10 (bottom half, right column). BM2 with Legendre polynomials and $b = 1$ achieves better results for V with respect to the bias for $K \geq 6$, but has high uncertainty. Here the marginal posterior distribution of σ_I, σ_ω and Algorithm 5 suggest, sensitively depending on κ^{max} and tol, $K = 5$ or $K = 8$ as optimal. However $K = 5$ is not optimal, as the posterior distribution of V is still biased and improves only with $K \geq 6$. With increasing K the uncertainty is increasing, which is also visible in the model

discrepancy posterior distribution in Figure 4.10 (top half), where the evolution of $\boldsymbol{\delta}^K$ for $K = 4, 5, 6, 11$ is illustrated. $\boldsymbol{\delta}^K$ for $K = 4$ roughly has the shape of the reference, but is, corresponding to the posterior distribution of V for $K = 4$, largely biased. This improves for $K = 5$ and gets even better for $K = 6$. With $K = 11$ first signs of overfitting start to occur, in particular for $\boldsymbol{\delta}_I^K$, where low frequency parts of the measurement noise are reproduced. This and the increased uncertainty are a sign for a too large K, i.e. respectively a too large polynomial degree. However, with increasing K the uncertainty does mainly grow in the coefficients of the constant and linear polynomial and not, as one might expect, in the coefficients of the higher degree polynomials. Consequently, the increasing uncertainty are signs of the identification problem, due to an growing number of unknowns with K.

As a theoretical experiment and since the reference model discrepancy is known, the influence of a more informative coefficient prior $\pi(\boldsymbol{a}) = Laplace(0, b)$ is tested by replacing the zero-mean for the coefficients of the constant polynomials $a_0^{(\cdot)}$ with an informative mean based on a polynomial fitted to the reference model discrepancy. All other coefficients $a_j^{(\cdot)}, j \geq 1$ remain with a zero-mean prior. Informative in $a_0^{(\cdot)}$ and still with $b = 1$ has almost no influence compared to previous results. But informative in $a_0^{(\cdot)}$ with $b = 0.1$ has an immense influence as displayed in Figure 4.9 (middle column). The results for V improve drastically with almost no bias and uncertainty. However, the availability of the used information about $a_0^{(\cdot)}$ is unrealistic in practice.

More realistic are general assumptions about the model discrepancy, for instance, that it is negligible in a stationary time domain. The reference model discrepancy for case J tends to zero for $t \to 6$. A model discrepancy prior for $\boldsymbol{\delta}^K(\boldsymbol{a})$ with the weighted Laguerre polynomials (scale $s = 6$), which also tends to zero for $t \to 6$ as displayed in Figure 4.3, consequently yields improved results. The posterior distributions of V in Figure 4.9 (right column) nicely converge to the reference value for increasing K. The marginal posterior distribution of σ_I, σ_ω and Algorithm 5 suggest $K = 4$ as optimal. With $K = 4$ the posterior distribution for $\boldsymbol{\delta}^K(\boldsymbol{a})$ perfectly infers the reference discrepancy with low uncertainty, see Figure 4.10 (bottom half). For $K \geq 7$ the uncertainty in V increases and for $K \geq 11$ the model discrepancy $\boldsymbol{\delta}^K(\boldsymbol{a})$ starts to overfit, too. Note that theoretically, for an optimal weighted Laguerre polynomial scaling parameter s, an even smaller K would be possible, since a stretched version of the model discrepancy for $K = 1$ could already fit the reference discrepancy well. However a non-optimal scale s works just fine as well, it only requires a larger K.

Inference of V and T simultaneously is displayed in Figure 4.11. The model discrepancy posterior distributions are displayed in the Appendix in Figure A.2. The BM1 results are good in V, but biased and overconfident in T. The GP results are uncertain and biased, even largely biased in T. The presented BM2 results are all with $b = 1$ for the coefficients prior $\pi(\boldsymbol{a})$. The results in the first column with Legendre polynomials are biased and uncertain, which repeats the observations previously made for cases zero

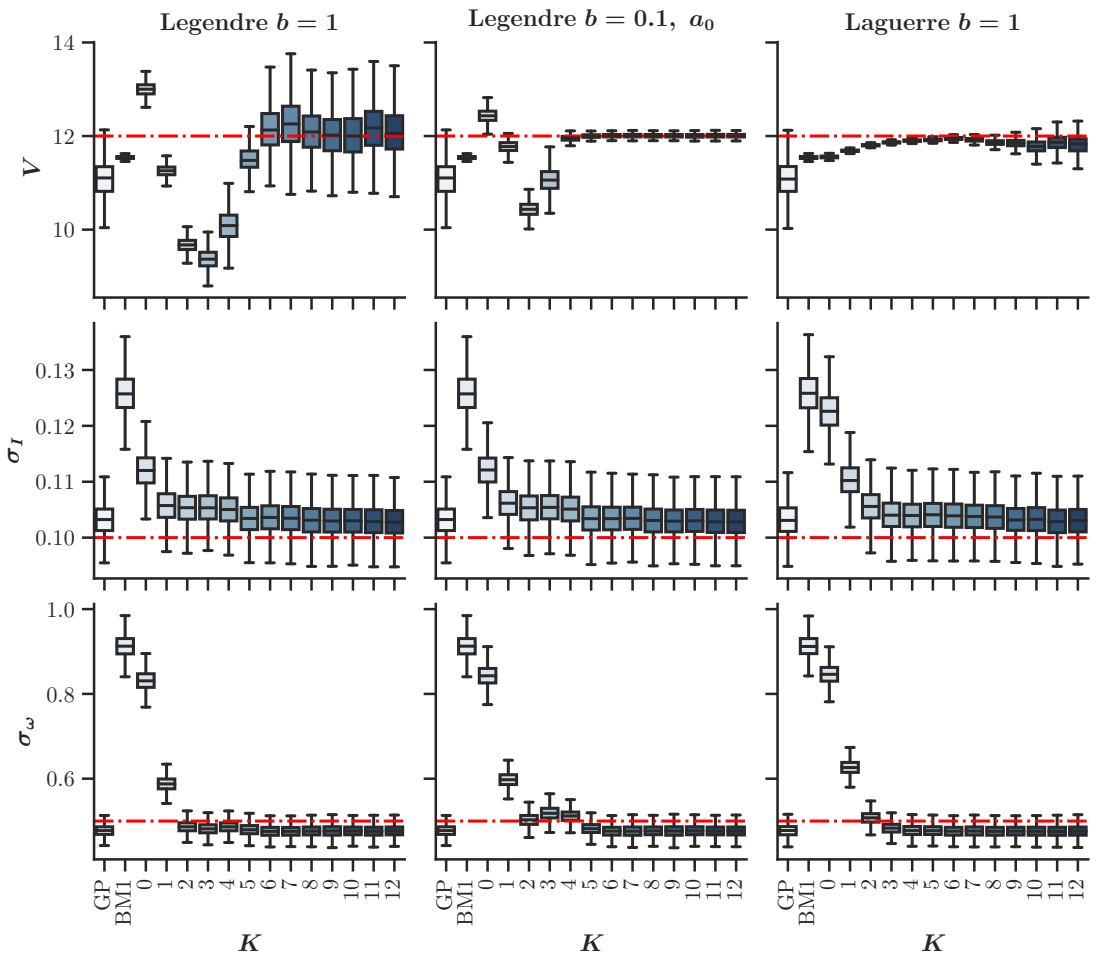


Figure 4.9.: Discrepancy case J posterior distributions for V, σ_I and σ_ω (rows). Results are obtained via GP, BM1 and BM2 for $K = 0, \ldots, 12$ with varying model discrepancy prior $\boldsymbol{\delta}^K(\boldsymbol{a})$ (columns), i.e. varying polynomials (Legendre and weighted Laguerre) and varying coefficient prior $\pi(\boldsymbol{a}) = Laplace(0, b)$ (middle column with informative mean for coefficients $a_0^{(\cdot)}$ based on the reference model discrepancy). Reference values are red dash-dotted.

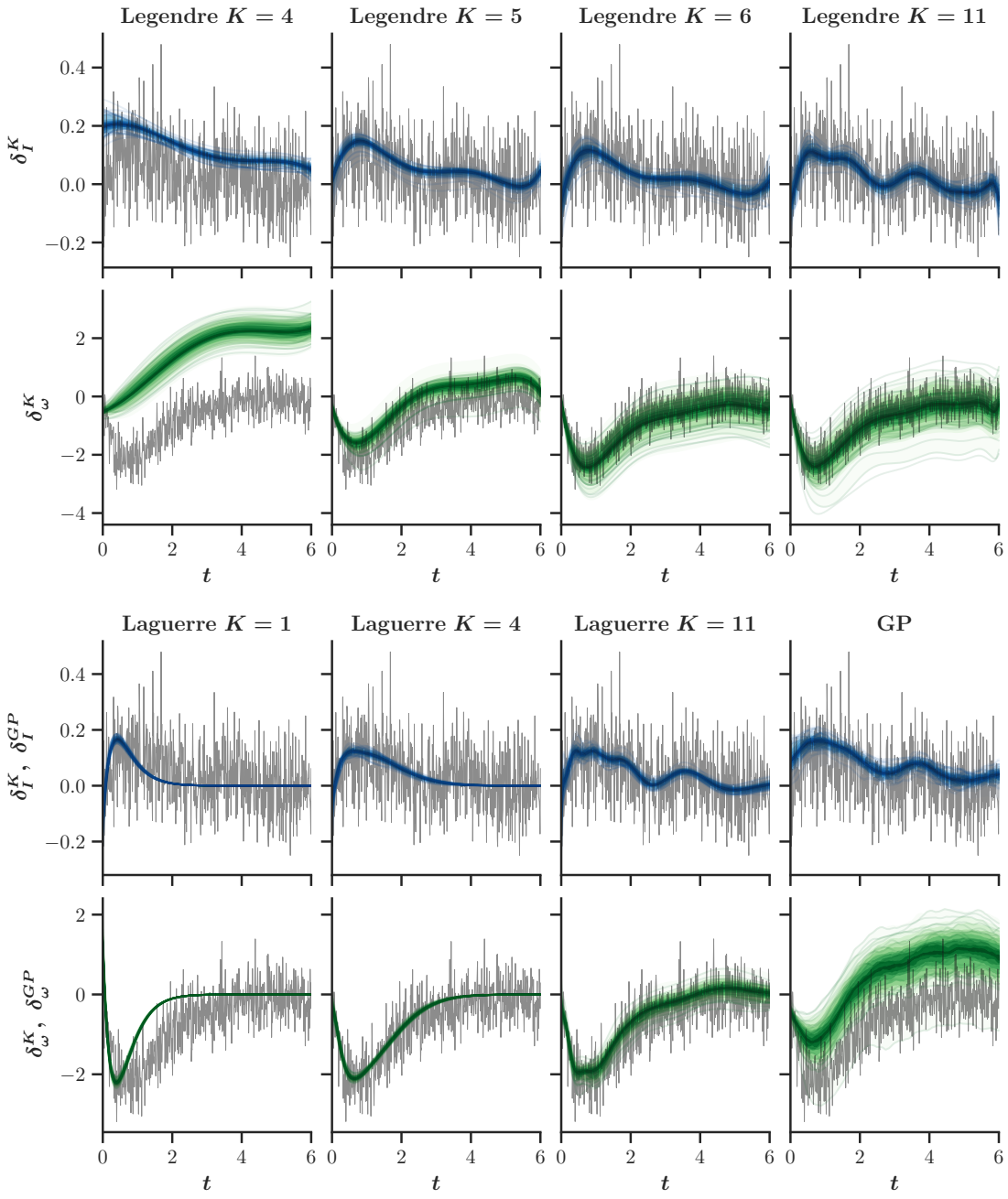


Figure 4.10.: Discrepancy case J posterior distributions of δ_I, δ_ω (rows) for inference of V, σ_I and σ_ω (see Figure 4.9). Results are obtained via BM2 with Legendre polynomials (top half), weighted Laguerre polynomials (bottom half) and K specified in the heading (all with $b = 1$) and GP (bottom half, right column). Reference discrepancy plus measurement noise is gray.

to quadratic. Algorithm 5 suggests an optimal $K = 6$. The corresponding model discrepancy posterior in Figure A.2 (1st column) is also uncertain and slightly biased, but with proper shape.

It is not included in the plot, but reducing b from 1 to 0.1 for the Legendre polynomials, as done previously, improves uncertainty and bias slightly. A way further and obvious improvement of the results is achieved by exchanging the default parameter prior with a more informative, centered prior $\pi_c(V, T) = \pi_c(V)\pi_c(T)$, with $\pi_c(V) = \mathcal{N}(12, 0.1^2)$ and $\pi_c(T) = \mathcal{N}(2.5, 0.1^2)$. This largely reduces bias and uncertainty (2nd column), however such specific prior information is hardly available in reality.

Again, BM2 with weighted Laguerre polynomials, representing more realistic information about the model discrepancy, leads to very good results (3rd column) for V and T with the default parameter prior. For an optimal $K = 4$ results correspond to the reference with only small uncertainty. Also the model discrepancy posterior is very sharp on the reference, see Figure A.2 (3rd column). This gets even better with $\pi_c(V, T)$ (4th column in Figure 4.11).

4.5.3. Test bench data

The test bench data and the corresponding simulation model are introduced in Section 2.2. For the following analysis the test bench run $\boldsymbol{y}_i$ with index $i = 15$ is used, denoted in the following by TB15. Parameter reference values are then the corresponding samples used for data generation, i.e. $V^\dagger = V_i$, $T^\dagger = T_i$. Reference values for σ_I, σ_ω are not available. Estimates based on Definition 1 with time interval $[3, 6]$ are $\overline{\sigma}_I = 0.19, \overline{\sigma}_\omega = 0.33$. But as they are only estimated for the last half of the whole time interval, they probably underestimate the actual noise standard deviations.

Inference results for V, T, σ_I and σ_ω with GP, BM1 and BM2 for $K = 0, \ldots, 18$ are displayed in Figure 4.12. The BM1 results are almost perfect for V, but biased for T. The GP results are largely biased in V and equal the prior in T. The BM2 results with Legendre polynomials also equal the prior in T and are largely biased in V, where the magnitude of the bias changes with K. The model discrepancy posterior distributions in Figure 4.13 (1st and 2nd column), somehow get the shape of the discrepancy, but are largely biased for $K = 6$ and tend to overfit for larger K, e.g. as displayed for $K = 18$. BM2 with weighted Laguerre polynomials achieves very good results in V for $K = 0, \ldots, 9$ and good results in T for $K = 1, 2, 3, 4, 6, 8$. For larger K bias and uncertainty increase. The model discrepancy posterior for $K = 5$ fits the discrepancy quite well. For $K = 18$ overfitting and bias is present, too.

The strong overfitting poses a challenge for selection of an optimal K with Algorithm 5, as the noise standard deviations still decrease for increasing K. With $\kappa^{max} = 1$ and a tolerant tolerance, $K = 2$ or $K = 6$ could be selected as optimal, both leading to decent model parameter estimates.

Additional results for test bench data with indices $i = 16, 26$, displayed in Figure A.3

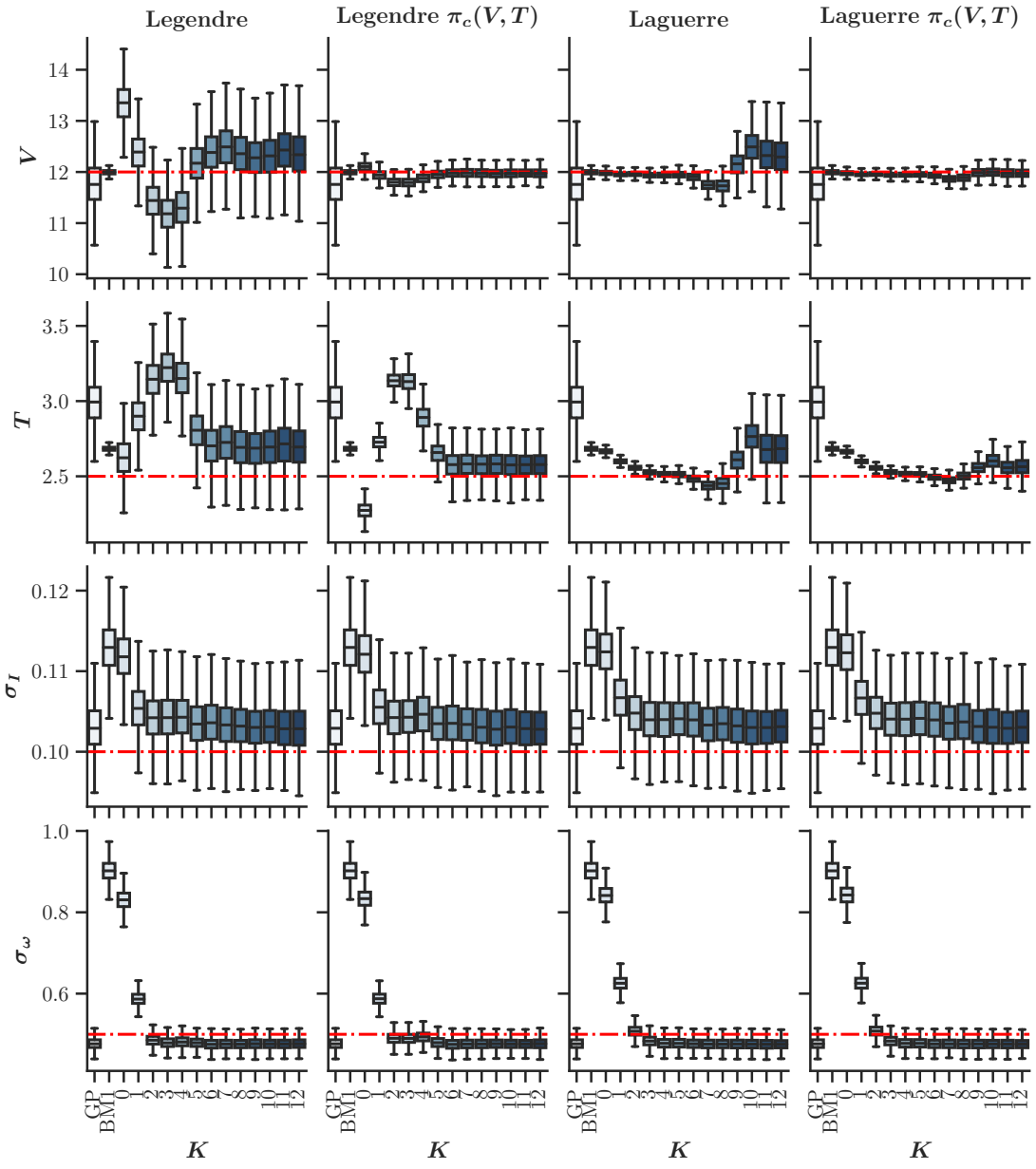


Figure 4.11.: Discrepancy case J posterior distributions for V, T, σ_I and σ_ω (rows). Results are obtained via GP, BM1 and BM2 for $K = 0, \ldots, 12$ with varying model discrepancy prior $\boldsymbol{\delta}^K(\boldsymbol{a})$ (columns), i.e. varying polynomials (Legendre and weighted Laguerre). The coefficient prior is default $\pi(\boldsymbol{a}) = Laplace(0, b = 1)$. 2nd and 4th column are with more informative, centered prior $\pi_c(V,T)$ for the parameters. Reference values are red dash-dotted.

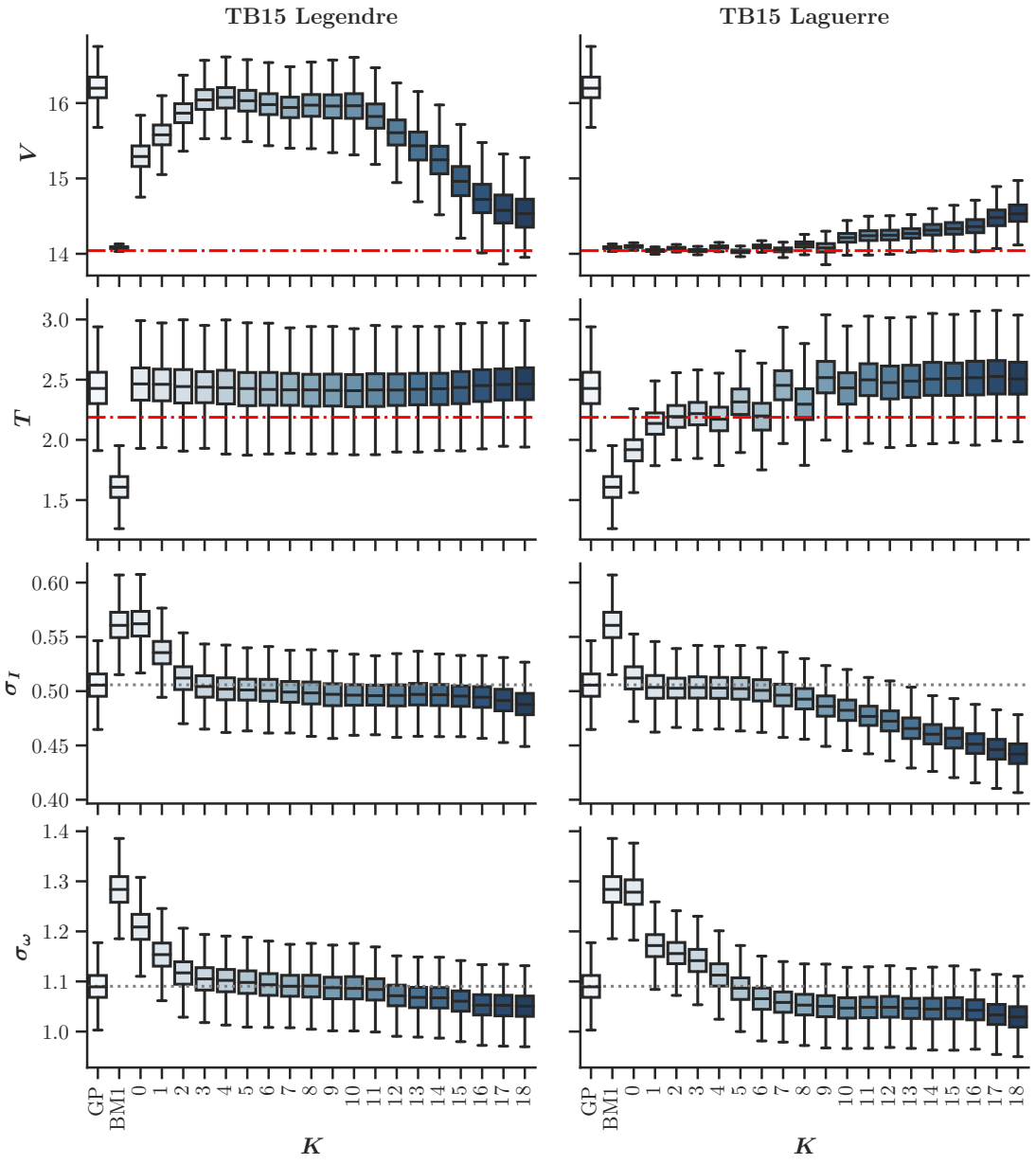


Figure 4.12.: Test bench data TB15 posterior distributions for V, T, σ_I and σ_ω (rows). Results are obtained via GP, BM1 and BM2 for $K = 0, \ldots, 18$ with varying model discrepancy prior $\boldsymbol{\delta}^K(\boldsymbol{a})$ (columns), i.e. Legendre and weighted Laguerre polynomials. The coefficient prior is default $\pi(\boldsymbol{a}) = Laplace(0, b = 1)$. Reference values are red dash-dotted. Dotted gray lines augment the GP noise standard deviation posterior mean for comparison.

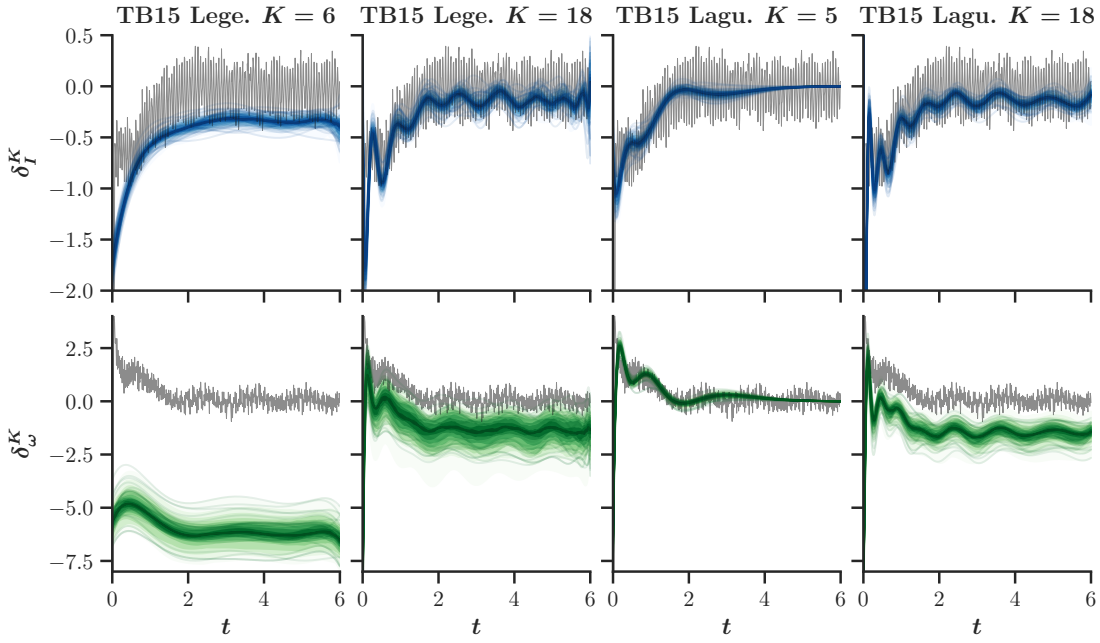


Figure 4.13.: Posterior distributions of δ_I, δ_ω (rows) for inference of V, T, σ_I and σ_ω (see Figure 4.12) for test bench data TB15. Results are obtained via BM2 with Legendre polynomials and weighted Laguerre polynomials with K specified in the heading (all with $b = 1$). Reference discrepancy plus measurement noise is gray.

show similar performance. The BM2 results with $K = 1, 2, 4$ for TB16 and with $K = 6, 9$ for TB26 show decent results, but selection of an optimal K with Algorithm 5 is again not obvious. For TB16 one might select $K = 4$ and for TB26 $K = 4$ or with small tolerance $K = 6$. The corresponding model discrepancy posterior distributions are displayed in Figure A.4.

4.5.4. Discussion

The inverse problems approached in this chapter are challenging, mainly due to the identification problem. The expectations on the introduced methods – BM2 and Algorithm 5 – are to improve on BM1 and GP solution and to get close to an unbiased parameter and model discrepancy estimation. At this point we would like to highlight again that the relatively simple function (polynomial) expansion with low number of terms for the model discrepancy in BM2 is chosen on purpose for three reasons: (1) It clearly improves on GP with respect to computational costs. (2) It has a low number of unknown parameters and (3) allows a certain smoothness which is together with (2) advantageous to tackle the identification problem. Of course, due to this modeling the BM2 model discrepancy has limited approximation capability. But it is better to obtain an approximation of the model discrepancy that might be rough, than none or only an highly uncertain and biased one because of unsolvable identifiability problems. This trade-off between solvability of the identifiability problem and approximation quality is visible in the numerical experiments above.

For the synthetic data, the BM1 results are generally biased and overconfident and the GP results are very uncertain. Overall, BM2 improves on this and achieves decent results, clearly satisfying the expectations. Note that some synthetic experiments show the difficulties of solving inverse problems with unknown model parameters and unknown model discrepancy, e.g. major identifiability problems occur for the joint inference of V and T for discrepancy types zero, constant, linear and quadratic. Nevertheless, Algorithm 5 worked well for selection of an optimal truncation parameter K for all examples, though it sensitively depends on the chosen tolerance tol and κ^{max} in some cases. Further, assuming additional knowledge about the model discrepancy, the weighted Laguerre polynomials are beneficial to reduce identifiability problems and lead to improved estimations with less uncertainty.

For the test bench data the results with BM2 and weighted Laguerre polynomials are decent as well and clearly satisfy expectations, e.g. for TB15 with $K = 5$ in Figure 4.12 and 4.13. However, selection of an optimal K with Algorithm 5 is difficult, as the BM2 model discrepancy term tends to overfit with increasing K.

The solution to this problem might be a stopping criterion for Algorithm 5 to prevent overfitting. This could be a hand tuned value or derived from the data for example as follows. As stated above in Section 4.3.2 Gaussian Processes automatically enforce model parsimony and avoid overfitting. Albeit the GP results for the model parameters

and the model discrepancy are biased, the overall data fit is still good. Consequently, the marginal posterior distribution of σ_I, σ_ω from the GP method might serve as orientation, where to stop in Algorithm 5. Consulting the synthetic results, it is confirmed to be a decent stopping criterion. For the test bench data, where the marginal posterior mean of σ_I, σ_ω from the GP method are visually extended (dotted gray) for comparison to the other methods, it serves well as a stopping criterion, too.

Of course one could make things simpler and fit a GP to the data directly without simulation model, but this would lead almost surely to a similar posterior for σ_I, σ_ω. To do so fit a GP with some kernel and i.i.d. noise to the data, until the desired data fit is achieved. Then, use the estimate of the noise standard deviation as stopping criterion. Doing so with an optimizer would require only a few covariance matrix inversions and result in acceptable computational costs.

Another idea is to accept the high computational costs of the GP method and improve the performance by transferring the information contained in the weighted Laguerre polynomials to the model discrepancy prior of the GP method. E.g. with a scaled covariance kernel $k_{scaled}(t, t') = \phi(t) k_{base}(t, t') \phi(t')$ for some function $\phi(t) \geq 0$ and a base kernel k_{base}. The amplitude of k_{scaled} then changes over time with ϕ and tends to zero if ϕ tends to zero. However, it is unclear if this leads to similar results as with BM2, and thus, further investigations are required in future work.

Chapter 5.

From deterministic to aleatoric parameter estimation

> Uncertainty is sometimes unavoidable. But in the world of scientific computing and engineering, at least, what's worse than uncertainty is being uncertain about how uncertain one is.
>
> DARPA 2015[1]

By moving to the stochastic inverse problem in Definition 5 we are now concerned with the estimation of aleatoric parameters in contrast to the inference of deterministic parameters in the inverse problem in Definition 4. The inference of aleatoric parameters is in principle an infinite dimensional problem, due to the unknown probability density functions that live in an infinite dimensional function space. This brings additional challenges, even higher computational costs and a model discrepancy that now depends on time and on parameters.

Part of theses challenges are addressed in this chapter by first introducing a discretization of the infinite dimensional problem with hierarchical estimation of aleatoric parameters in Section 5.1. Following, Section 5.1.1 presents posterior distribution approximation with a hierarchical surrogate-based MCMC method and Section 5.1.2 proposes a novel surrogate-based ABC method with summary statistics that improves on MCMC with respect to computational costs. This is accompanied by numerical results for synthetic and test bench data in Section 5.2.

Additional consideration of model discrepancy will be covered later in Chapter 6.

5.1. Hierarchical aleatoric parameter estimation

The need for hierarchical models in Bayesian statistics and in particular in the context of non-parametric methods (i.e. methods in function space) in machine learning, is well established [Bishop, 2006]. For state of the art in hierarchical Bayesian estimation we refer to [Robert, 2007] and for hierarchical Bayesian inverse problems to [Sraj et al.,

[1] Defense Advanced Research Projects Agency (DARPA) see https://www.darpa.mil/news-events/2015-01-08

2016, Dunlop et al., 2017, Roininen et al., 2019, Latz et al., 2019] and the references therein. Those methods deal with inference of continuous-parameter random fields both for priors and hyper-priors.

As this is an active research field in its own, we assume in the following parametric distributions for the unknown aleatoric parameters and thus constrain, i.e. discretize the function space. In particular, we assume Gaussian distributions and inference is then based on the hyper-parameters mean and standard deviation, following [Glaser et al., 2016, Glaser et al., 2017, Glaser, 2020]. This corresponds to the traditional way of dimensionality reduction in forward and inverse stochastic problems by using the truncated Karhunen-Loève expansion (KLE) [Marzouk and Najm, 2009, Le Maître and Knio, 2010, Sraj et al., 2016], with only two coefficients.

For theoretical error analysis, i.e. whether the reduced problem can well approximate the original one, we refer to [Li, 2015]. There, a proof shows that the maximum a posteriori (MAP) estimator is well approximated by the truncated KLE. Further several works on theoretical error analysis in the forward problem are mentioned. The effects of the truncated KLE on the Bayesian inverse problem solution is investigated in [Uribe et al., 2020].

In the remainder of this section we formulate the stochastic inverse problem in a hierarchical way and present two methods – hierarchical MCMC and hierarchical ABC – to approximate the posterior distribution. For convenience we already use the notation of the electric motor application. Note that the following text is based on the authors own work [John et al., 2020].

Recall the problem description of the stochastic inverse problem in Definition 5, with data $\mathcal{Y}$, simulation model $\mathcal{M}(X)$ and the unknown aleatoric parameters X with unknown probability distribution $\pi(X)$. The approaches presented so far in Chapters 3 and 4 are for inference of unknown deterministic parameters $\boldsymbol{x} \in \mathbb{R}^n$. A prior distribution reflects the a-priori belief about possible values of $\boldsymbol{x}$ before data is observed. This prior is then updated in the Bayesian inference scheme to obtain a posterior distribution. Now again, to make things clear, this prior and posterior distribution only reflect the degree of belief on what values are plausible for $\boldsymbol{x}$, and do not reflect that $\boldsymbol{x}$ is a random variable (or a random vector). Thus, in order to estimate the unknown probability distribution $\pi(X)$, it is factorized and hierarchically approximated by

$$\pi(X \mid \boldsymbol{\theta}) = \prod_{i=1}^{n} \pi(X_i \mid \boldsymbol{\theta}), \tag{5.1}$$

where $\boldsymbol{\theta}$ is a real-valued vector of unknown hyper-parameters for the parametric distributions. For $X = (V, T)$, we assume $\pi(V \mid m_V, \sigma_V) = \mathcal{N}(m_V, \sigma_V^2)$ and $\pi(T \mid m_T, \sigma_T) = \mathcal{N}(m_T, \sigma_T^2)$ where $m_V, m_T \in \mathbb{R}$, $\sigma_V, \sigma_T > 0$, hence, $\boldsymbol{\theta} := (m_V, m_T, \sigma_V, \sigma_T)$. Let $\pi_0(\boldsymbol{\theta})$ denote the prior for the hyper-parameters. As the distribution of X is now fully determined by the hyper-parameters $\boldsymbol{\theta}$, the goal is to approximate $\pi(\boldsymbol{\theta} \mid \mathcal{Y})$ instead of $\pi(X \mid \mathcal{Y})$.

Let $\boldsymbol{y}_i^{(\cdot)} \in \mathcal{Y}$ be one of the measurements, where $(\cdot)$ either stands for the current I or angular velocity ω. Let $\boldsymbol{x}_i = X(\omega)$ denote a realization of the random vector X. Further, let $\mathcal{M}^{(\cdot)}(\boldsymbol{x}_i)$ denote the output of the simulation for model parameters $\boldsymbol{x}_i$ either for I or ω. We assume additive Gaussian measurement noise that is independent and identically distributed

$$\begin{aligned} \boldsymbol{y}_i^{(\cdot)} = & \underbrace{\mathcal{M}^{(\cdot)}(\boldsymbol{x}_i) + \boldsymbol{\varepsilon}_i}_{=: \mathcal{G}^{(\cdot)}(\boldsymbol{x}_i, \sigma_i^{(\cdot)})}, \quad \boldsymbol{\varepsilon}_i \sim \mathcal{N}(0, \Sigma_i^{(\cdot)}), \quad \Sigma_i^{(\cdot)} = \sigma_i^{(\cdot)} \mathbf{I}_{N_t}, \end{aligned} \tag{5.2}$$

with $\sigma_i^{(\cdot)} > 0$ for $i = 1, \ldots, N$. Here $\mathcal{G}^{(\cdot)}(\boldsymbol{x}_i, \sigma_i^{(\cdot)})$ denotes the generative model that produces noisy simulations for given $\boldsymbol{x}_i, \sigma_i^{(\cdot)}$.

As already motivated in Section 3.3.1 we replace the original simulation model $\mathcal{M}(X)$ by a PCE surrogate $\mathcal{M}^{PCE}(X)$. For surrogate generation we assume $\pi(X)$ to be uniform within a parameter domain $\mathcal{X}^{PCE} \subset \mathbb{R}^n$ with $\mathcal{X}^{PCE}$ large enough to cover the whole range of the hierarchical prior distribution $\pi(X \mid \boldsymbol{\theta})\pi_0(\boldsymbol{\theta})$.

At this point the hierarchical MCMC and the hierarchical ABC method deviate in approximating the posterior distribution.

5.1.1. Hierarchical surrogate-based MCMC

Picking up from the noise model (5.2) the likelihood for one measurement is then given as

$$\pi(\boldsymbol{y}_i^{(\cdot)} \mid \boldsymbol{x}_i, \sigma_i^{(\cdot)}) = \mathcal{N}\left(\boldsymbol{y}_i^{(\cdot)} - \mathcal{M}^{(\cdot)}(\boldsymbol{x}_i), \Sigma_i^{(\cdot)}\right). \tag{5.3}$$

And the likelihood for all measurements in $\mathcal{Y}$ as

$$\pi(\mathcal{Y} \mid X, \boldsymbol{\sigma}) = \prod_{i=1}^{N} \mathcal{N}\left(\boldsymbol{y}_i^I - \mathcal{M}^I(\boldsymbol{x}_i), \Sigma_i^I\right) \mathcal{N}\left(\boldsymbol{y}_i^\omega - \mathcal{M}^\omega(\boldsymbol{x}_i), \Sigma_i^\omega\right). \tag{5.4}$$

Here $\boldsymbol{\sigma}$ denotes the vector of all $\sigma_i^{(\cdot)}$ and $\pi_0(\boldsymbol{\sigma})$ the corresponding prior. The posterior distribution is

$$\pi(\boldsymbol{\theta}, \boldsymbol{\sigma} \mid \mathcal{Y}) \propto \pi(\mathcal{Y} \mid X, \boldsymbol{\sigma}) \pi(X \mid \boldsymbol{\theta}) \pi_0(\boldsymbol{\theta}) \pi_0(\boldsymbol{\sigma}). \tag{5.5}$$

To approximate the posterior distribution $\pi(\boldsymbol{\theta}, \boldsymbol{\sigma} \mid \mathcal{Y})$ with MH-MCMC (see Algorithm 1) in the hierarchical setup one needs to sample the following parameters: the hyper-parameters $\boldsymbol{\theta}$, the measurement noise standard deviations $\boldsymbol{\sigma}$, and the realizations $\boldsymbol{x}_i = X(\omega), i = 1, \ldots, N$. For $X = (V, T)$, this is a total of $4 + 2N + 2N$ parameters to sample from. For a visualization of the inference structure see the graphical model[2] in Figure 5.1.

[2] For details on graphical models we refer to [Bishop, 2006, Ch. 8].

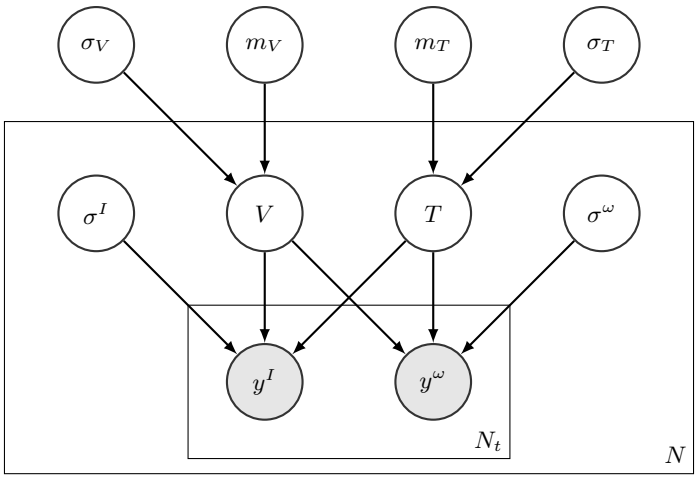


Figure 5.1.: Graphical model of the hierarchical inference structure for MCMC.

As stated in Section 3.1.2 the sampling might be highly inefficient in the case of concentrated posterior distributions due to highly informative data. Handing over MAP estimates for initialization and the inverse of the Hessian is an option to increase sampler efficiency. However computing the full inverse Hessian is to expensive, thus we suggest to estimate only the MAP and the inverse Hessian for the realizations $\boldsymbol{x}_i = X(\omega), i = 1, \ldots, N$ by

$$(\boldsymbol{x}_i)_{\text{MAP}} = \arg\min_{\boldsymbol{x}_i \in \mathcal{X}} \|\boldsymbol{y}_i^I - \mathcal{M}^I(\boldsymbol{x}_i)\|_{\Sigma_i^I} + \|\boldsymbol{y}_i^\omega - \mathcal{M}^\omega(\boldsymbol{x}_i)\|_{\Sigma_i^\omega} + log(\pi(X)). \tag{5.6}$$

And then use the $(\boldsymbol{x}_i)_{\text{MAP}}$ for initialization and only a diagonal approximation $\text{diag}(\tilde{C})$ of C for the proposal distribution.

In principle, sampling efficiency can be improved with an advanced gradient-based sampler and a surrogate-based gradient, as introduced in Section 4.4. A remedy are the increasing computational costs, due to the additional gradient evaluations. This poses major drawbacks for the hierarchical approach in particular, where each sample requires N simulation model evaluations and, consequently, with gradient-based methods N additional gradient evaluations. Gradient-based sampling is thus not considered in this chapter. MH-MCMC serves well as a baseline and for comparison to the ABC method suggested in the following section.

5.1.2. Hierarchical surrogate-based ABC with summary statistics

Again, picking up from the noise model (5.2) we use summary statistics to summarize the data $\mathcal{Y}$ and the generative model $\mathcal{G}^{(\cdot)}(X, \sigma_i^{(\cdot)})$, for I and ω respectively. For an introduction to ABC methods and summary statistics see Section 3.1.3. In general one

could use any summary statistic that seems useful for the given data. Here we use the empirical mean and standard deviation, i.e.

$$S_1^{(\cdot)} := \frac{1}{N}\sum_{i=1}^{N} \boldsymbol{y}_i^{(\cdot)} \quad \text{and} \quad S_2^{(\cdot)} := \sqrt{\frac{1}{N-1}\sum_{i=1}^{N}(\boldsymbol{y}_i^{(\cdot)} - S_1^{(\cdot)})^2}. \tag{5.7}$$

The summary of the data $\mathcal{Y}$ is then

$$S(\mathcal{Y}) := (S_1^I,\ S_1^\omega,\ S_2^I,\ S_2^\omega) \in \mathbb{R}^{4N_t}. \tag{5.8}$$

At this point we could do so with the generative model as well: generate a set of noisy simulation data $\mathcal{Y}^{\mathcal{G}}$ depending on samples of $X \mid \boldsymbol{\theta}$ for given $\boldsymbol{\theta}$ and $\sigma_i^{(\cdot)}$, then compute $S(\mathcal{Y}^{\mathcal{G}})$ and compare to $S(\mathcal{Y})$. The size $N^{\mathcal{G}}$ of $\mathcal{Y}^{\mathcal{G}}$ does not necessarily need to correspond to the size N of $\mathcal{Y}$ as only the summary statistics are compared. However $N^{\mathcal{G}}$ needs to be sufficiently large to obtain accurate summary statistics. Considering the slow convergence rate of Monte Carlo integration, improving the quality of $S(\mathcal{Y}^{\mathcal{G}})$ by increasing $N^{\mathcal{G}}$ might get prohibitively expensive as $S(\mathcal{Y}^{\mathcal{G}})$ needs to be computed for each sample ($S(\mathcal{Y})$ only once!).

In the following we introduce a more efficient approach by exploiting the additive noise structure and the parametric distribution of X further. Assume $\sigma_i^{(\cdot)} = \sigma^{(\cdot)}$ for $i = 1, \dots, N$. This assumption makes sense since summarizing the data with mean and standard deviation makes it anyway impossible to infer the noise structure of individual measurements $\boldsymbol{y}_i^{(\cdot)}$. Then the mean of the generative model with respect to $\pi(X \mid \boldsymbol{\theta})$ is

$$S_1^{(\cdot)}(\boldsymbol{\theta}) := \mathbb{E}[\mathcal{G}^{(\cdot)}(X, \sigma^{(\cdot)})] = \mathbb{E}[\mathcal{M}(X) + \boldsymbol{\varepsilon}^{(\cdot)})] = \mathbb{E}[\mathcal{M}(X)] + \underbrace{\mathbb{E}[\boldsymbol{\varepsilon}^{(\cdot)})]}_{=0} \tag{5.9}$$

$$= \int_{\mathcal{X}} \mathcal{M}(\boldsymbol{x})\pi(\boldsymbol{x} \mid \boldsymbol{\theta})d\boldsymbol{x}. \tag{5.10}$$

Since noise $\boldsymbol{\varepsilon}$ is assumed to be stochastic independent and Gaussian distributed with zero mean, it cancels by taking the expectation. For the standard deviation we obtain

$$S_2^{(\cdot)}(\boldsymbol{\theta}, \sigma^{(\cdot)})^2 := V[\mathcal{G}^{(\cdot)}(X, \sigma^{(\cdot)})] = V[\mathcal{M}(X) + \boldsymbol{\varepsilon}^{(\cdot)})] = V[\mathcal{M}(X)] + \underbrace{V[\boldsymbol{\varepsilon}^{(\cdot)})]}_{=\mathrm{diag}(\Sigma^{(\cdot)})}$$

$$= \int_{\mathcal{X}} \left(\mathcal{M}(\boldsymbol{x}) - S_1^{(\cdot)}(\boldsymbol{\theta})\right)^2 \pi(\boldsymbol{x} \mid \boldsymbol{\theta})d\boldsymbol{x} + \mathrm{diag}(\Sigma^{(\cdot)}). \tag{5.11}$$

In order to approximate the multidimensional integrals $S_1^{(\cdot)}(\boldsymbol{\theta})$ and $S_2^{(\cdot)}(\boldsymbol{\theta}, \sigma^{(\cdot)})$ one can either do Monte Carlo sampling or exploit the assumption of the parametric distribution $\pi(X \mid \boldsymbol{\theta})$ further and utilize Sparse Grid quadrature $\mathcal{A}(L^{SG}, \mathcal{Q})$ from (3.32)

with respect to $\pi(X \mid \boldsymbol{\theta})$, for a Sparse Grid level $L^{SG} \in \mathbb{N}_0$. Sparse Grid quadrature decreases the number of simulation model evaluations (respectively surrogate evaluations) further by simultaneously increasing approximation quality. We denote the Sparse Grid approximations of the summary statistics integrals by

$$\hat{S}_1^{(\cdot)}(\boldsymbol{\theta}; L^{SG}) = \mathcal{A}(L^{SG}, \mathcal{Q})(\mathcal{M}), \tag{5.12}$$

$$\hat{S}_2^{(\cdot)}(\boldsymbol{\theta}, \sigma^{(\cdot)}; L^{SG}) = \mathcal{A}(L^{SG}, \mathcal{Q})\left(\left(\mathcal{M} - \hat{S}_1^{(\cdot)}(\boldsymbol{\theta}; L^{SG})\right)^2\right) + (\sigma^{(\cdot)})^2, \tag{5.13}$$

where L^{SG} denotes the dependence on the Sparse Grid level and corresponding approximation quality. The summary of the generative model is then

$$S(\boldsymbol{\theta}, \sigma^I, \sigma^\omega; L^{SG}) := (\hat{S}_1^I(\boldsymbol{\theta}; L^{SG}),\ \hat{S}_1^\omega(\boldsymbol{\theta}; L^{SG}),\ \hat{S}_2^I(\boldsymbol{\theta}, \sigma^I; L^{SG}),\ \hat{S}_2^\omega(\boldsymbol{\theta}, \sigma^\omega; L^{SG})) \in \mathbb{R}^{4N_t}. \tag{5.14}$$

As the current and the angular velocity have different magnitude it is important to normalize the summary statistics. To do so we divide each component in $S(\mathcal{Y})$ by its L^1 norm, i.e. $\frac{S_{(\cdot)}}{\|S_{(\cdot)}\|_1}$ and so on, and use the exact same scaling also for the components of $S(\boldsymbol{\theta}, \sigma^I, \sigma^\omega; L^{SG})$.

Let $d(\cdot, \cdot)$ be the L^2 norm. Note that this particular choice of the distance plus the mean and standard deviation as summary statistics is related to the 2-Wasserstein metric for comparison of two Gaussian Processes[3]. See [Mallasto and Feragen, 2017] and the references therein for further details on the 2-Wasserstein metric for GPs.

Finally, for a given threshold $\delta > 0$ and priors $\pi_0(\boldsymbol{\theta}), \pi_0(\sigma)$, we sample from the posterior $\pi(\boldsymbol{\theta}, \sigma^I, \sigma^\omega \mid \mathcal{Y})$ or rather $\pi(\boldsymbol{\theta}, \sigma^I, \sigma^\omega \mid S(\mathcal{Y}))$ with Algorithm 3 or Algorithm 4 and the normalized summary statistics $S(\mathcal{Y})$ and $S(\boldsymbol{\theta}, \sigma^I, \sigma^\omega; L^{SG})$. For a visualization of the inference structure see the graphical model in Figure 5.2.

The noise standard deviations σ^I and σ^ω are difficult to infer with the chosen summary statistics, as they are only additive terms in $S_2^{(\cdot)}(\boldsymbol{\theta}, \sigma^{(\cdot)})$. Thus we estimate them a-priori and keep them fixed during inference (Another option would be to define a highly informative prior centered on the a-priori estimation, to allow the sampler to deviate a bit from the fixed value).

We estimate the noise standard deviation of each measurement $\boldsymbol{y}_i$ in a stationary time interval according to Definition 1 and denote the estimates by $\overline{\sigma_i^{(\cdot)}}$ for $i = 1, \ldots, N$. Then we use either their mean or median as estimate for σ^I and σ^ω.

Remark. Note that similar to the MCMC case one could use MAP $\boldsymbol{\theta}_{MAP}$ and inverse Hessian estimates C of the parameters to accelerate the ABC sampler efficiency, e.g. by

[3] For two Gaussian measures $\nu_1 = \mathcal{N}(m_1, C_1), \nu_2 = \mathcal{N}(m_2, C_2)$ in $\mathbb{R}^{N_t}$ the 2-Wasserstein metric is $W_2(\nu_1, \nu_2) = \|m_1 - m_2\|_2^2 + \|C_1^{1/2} - C_2^{1/2}\|_F^2$. For C_1, C_2 diagonal, the Frobenius norm simplifies to the L^2 norm of the standard deviations $\|diag(C_1^{1/2}) - diag(C_2^{1/2})\|_2^2$.

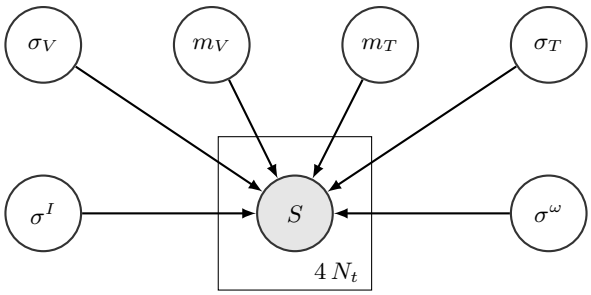


Figure 5.2.: Graphical model of the hierarchical inference structure for ABC.

modifying the prior to $\pi_0(\boldsymbol{\theta}) = \mathcal{N}(\boldsymbol{\theta}_{MAP}, sC)$ for $s > 1$. Provided that the estimates are correct this would lead to less rejected samples. But assume these estimates are distorted, then the modified prior (in case it is more informative than the original) might prevent us from sampling in regions where the posterior is non-zero, leading to a biased posterior. Consequently, in this work we are not using MAP and inverse Hessian estimates for ABC.

5.2. Numerical experiments

For the numerical experiments in this section the method *hierarchical surrogate-based MCMC* summarized in Section 5.1.1 is abbreviated by MCMC and in the case of initialization with MAP estimates by MCMC(MAP). Further, the method *hierarchical surrogate-based ABC with summary statistics* introduced in Section 5.1.2 is abbreviated by ABC and in the case of SMC by SMC ABC.

The numerical experiments are first presented for the basic electric motor model (see Section 2.1) with synthetic data and second for the test bench data with corresponding complex model (2.12) (see Section 2.2). Section 5.2.1 details on the joint inference setup. Then Section 5.2.2 and Section 5.2.3 present the numerical results for synthetic and test bench data, respectively. Section 5.2.4 concludes with a brief discussion of the results. Note that the following is restated from the authors own work [John et al., 2020] with slight modifications.

5.2.1. Inference setup

In both cases noisy measurement data $\mathcal{Y}$ for the electric current I and angular velocity ω is given to infer the distributions of parameters voltage V and load torque T. In Section 5.1 both are modeled as Gaussian random variables with unknown hyper-parameters mean m_v, m_T and standard deviations σ_V, σ_T. The prior distributions for m_V, m_T are uniform with bounds $-30\%/+50\%$ based on the reference values and for σ_V, σ_T with

bounds $-75\%/+125\%$ based on the reference values. For MCMC the prior distributions for σ^I and σ^ω are inverse Gamma distributions with mean and standard deviation 0.1. Due to the summary statistics, the identifiability of the noise is difficult. Thus, estimates of σ^I and σ^ω based on the available data and Equation (2.6) are used for ABC.

For ABC a fixed threshold $\delta > 0$ and number of samples $m = 1500$ are used, resulting in M proposals that are determined by the algorithm. Similar for SMC ABC, where a fixed, decreasing set of tolerance thresholds and $m = 1500$ determine M. For MCMC and MCMC(MAP) three parallel Markov chains, each of length 11.000, are sampled. With a burn-in phase of 6.000 samples and a modest thinning (discard every second sample), due to autocorrelation, three times 2.500 samples are then used to approximate the posterior distribution.

For ABC and SMC ABC an implementation based on the Python package ELFI [Lintusaari et al., 2018] is used and for MH-MCMC an implementation based on the Python package PyMC3 [Salvatier et al., 2016], with Gaussian random walk proposal. All computations are performed on a standard work station. Modest multiprocessing on 3 cores is used for all methods, i.e. for MCMC three parallel Markov Chains are generated and for ABC sampling is also carried out in parallel.

5.2.2. Synthetic data

In the following we first describe the synthetic data generation, comment on the PCE surrogate, present detailed numerical results and finally draw conclusions on posterior consistency.

Synthetic data generation

To generate synthetic measurement data $\mathcal{Y}$, the generative model $\mathcal{G}(X, \sigma) = \mathcal{M}(X) + e$ is used, where the simulation model $\mathcal{M}$ is the basic electric motor model (2.3). Reference distributions for the aleatoric parameters $X = (V, T)$ are defined by $\pi(V \mid m_V, \sigma_V) = \mathcal{N}(m_V, \sigma_V^2)$ and $\pi(T \mid m_T, \sigma_T) = \mathcal{N}(m_T, \sigma_T^2)$, with hyper-parameters $m_V = 12, m_T = 2.5$, $\sigma_V = 0.7, \sigma_T = 0.2$. Further, reference values for the noise standard deviations are $\sigma^I = 0.1$ and $\sigma^\omega = 0.5$. With this setting, we sample N times from $\mathcal{G}(X, \sigma)$ in order to generate the synthetic measurement data $\mathcal{Y}$. Basically, the simulation model $\mathcal{M}$ is evaluated at N independent samples of X, resulting in N model outputs containing discrete time series of current I and rotational speed ω each of size N_t. Then for each output independent Gaussian noise is added according to e^I and e^ω. Figure 5.3 shows the reference distributions, the samples and Gaussian distributions fitted to the samples for V and T, respectively. An overview of the resulting data $\mathcal{Y}$ for $N = 100$ is visualized in Figure 5.4. Later, Figure 5.7 displays a histogram, mean and median of the noise standard deviation estimations $\overline{\sigma_i^{(\cdot)}}, i = 1, \ldots, N$ based on Equation (2.6). To obtain a correct estimate of σ_V and σ_T it is important to estimate the noise standard deviations

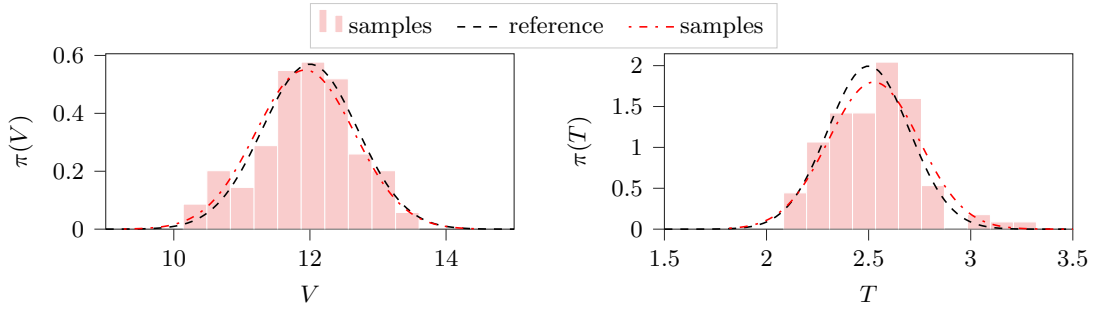


Figure 5.3.: The figures show the Gaussian reference distributions (black dashed) of the parameters V and T, histograms of $N = 100$ samples and Gaussian distributions fitted to the samples (red dash-dotted).

correctly. An underestimation of the noise standard deviation results in an overestimation of σ_V or σ_T and vice versa, this is due to the additive structure in the summary statistic, see Equation (5.11).

PCE surrogate

A PCE surrogate as introduced in Section 3.3.1 is used. Validation is carried out with a set of 100 random samples and corresponding simulations. The RMSE scaled by the standard deviation of the validation set for a level $L = 2$ PCE (leads to 17 Sparse Grid points) is in the range of 10^{-7}. The surrogate for this example can be seen as almost exact. The speed up is approximately factor 100. I.e. evaluation of one sample with the original model $\mathcal{M}$ takes $4.5ms \pm 0.412ms$ and with the surrogate $\mathcal{M}^{PCE}$ only $0.0446ms \pm 0.007ms$ (mean $\pm$ std. dev. of 7000 runs). Further the surrogate model, in particular the polynomial evaluation, can be vectorized to evaluate a batch of samples simultaneously, which increases efficiency additionally.

Results

Figure 5.5 displays boxplots of the samples approximating the marginal posterior distributions of the hyper-parameters, obtained with ABC, SMC ABC, MCMC and MCMC(MAP). Adding on this Table 5.1 lists the corresponding runtimes, number of proposals and samples of all methods. For ABC four decreasing values of $\delta \in \{10, 7.5, 5, 2.5\}$ show the concentration of the marginal posterior. The number M of ABC sample proposals to obtain a fixed number of $m = 1500$ samples that satisfy $d(\cdot, \cdot) \leq \delta$ increases exponentially, as expected. The SMC ABC results are obtained with the exact same δ's and are comparable to the ABC results for the smallest δ. However, by reusing the information of sample-discrepancy pairs from previous populations, SMC

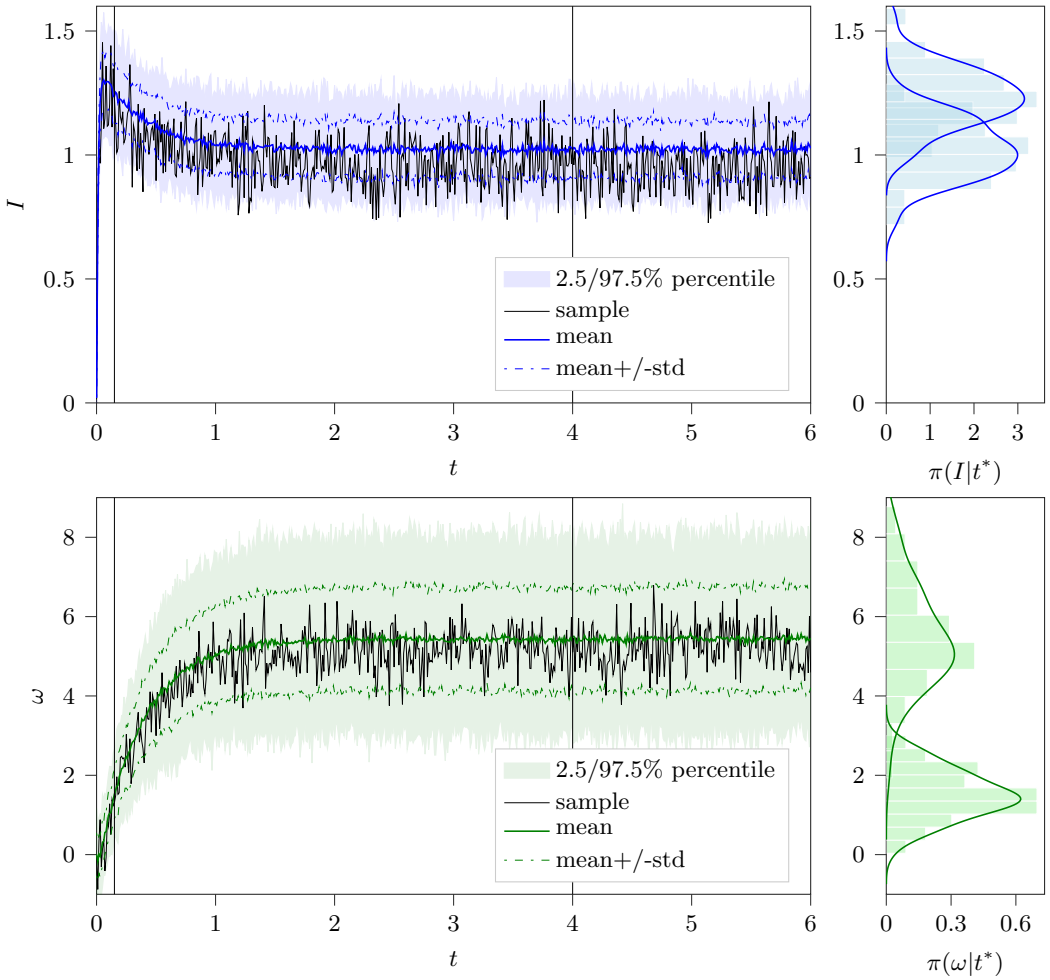


Figure 5.4.: The figures show the artificially generated noisy measurements $\mathcal{Y}$ of the current I and the rotational speed ω for $N = 100$. The area between the 2.5% and 97.5% percentile (shaded), mean+/-standard deviation (dash-dotted) and the mean (solid) of all N measurements are depicted. The black lines show an exemplary noisy sample measurement series. Further, at two time points (vertical lines at $t^* = 0.15$ and $t^* = 4$ seconds) histograms and kernel density plots are displayed on the right hand side.

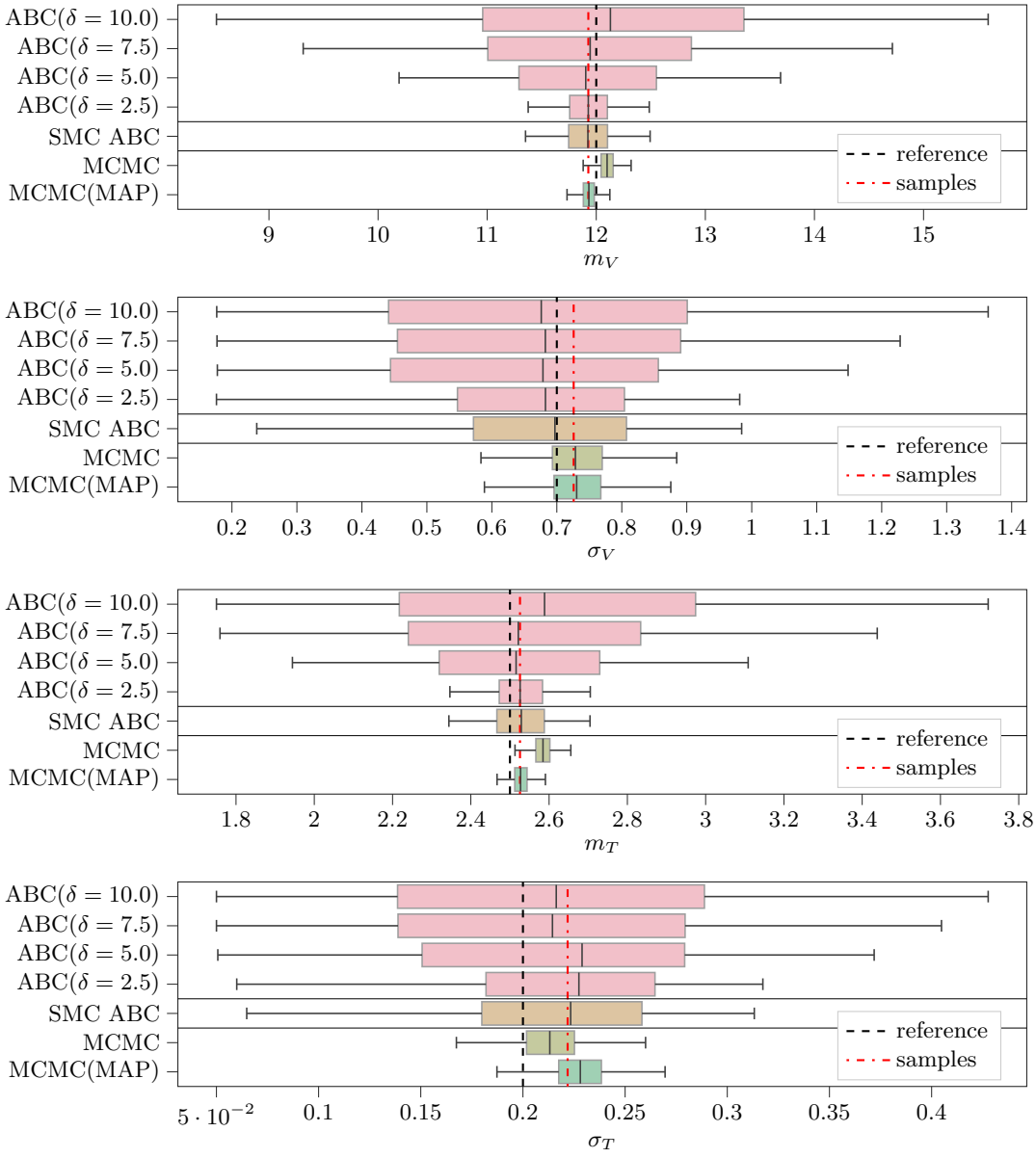


Figure 5.5.: Boxplots of the samples approximating the hyper-parameters marginal posterior distributions in the synthetic case obtained with ABC, SMC ABC, MCMC and MCMC(MAP) in comparison to the reference values and sample mean and standard deviation.

method	runtime [s]	samples m	proposals M
ABC($\delta = 10.0$)	3	1500	12000
ABC($\delta = 7.5$)	7	1500	28000
ABC($\delta = 5.0$)	27	1500	106000
ABC($\delta = 2.5$)	1536	1500	5718000
SMC ABC	211	1500	626000
MCMC	400	n_{eff}	3*11000
MCMC(MAP)	387	n_{eff}	3*11000

Table 5.1.: Runtimes in seconds, number of samples m and proposals M of the considered methods. For MCMC the number of effective samples n_{eff} is given in detail in Table 5.2.

ABC reduces the overall number of proposals by approximately factor 10 in order to obtain a similar result.

Table 5.2 lists for MCMC and MCMC(MAP) the Gelman-rubin statistics $\hat{R}$ and the number of effective samples n_{eff} of the samples (without thinning). Optimal sampling efficiency would require $\hat{R} = 1$ and n_{eff} close to the number of considered samples, see [Gelman et al., 2013] for further details. The statistics show that MCMC has clearly difficulties to converge and an increase of sample size would be recommended. It is a bit better for the hyper-parameters and noise standard deviations, but poor for V_i and T_i for $i = 1, \ldots, N$, which is due to the concentration effect of the posterior and the high dimensions. Overall, this sampling difficulties lead to a biased posterior for m_V and m_T, displayed in Figure 5.5. By initializing with MAP estimates MCMC(MAP) shows clear improvements in sampler statistics and posterior distribution. However the sampler has still difficulties, e.g. the number of effective samples is still less than 15% of the considered samples.

In addition to the posterior distributions of the hyper-parameters, marginal posterior distributions of V_i and T_i for $i = 1, \ldots, N$ obtained via MCMC(MAP) are visualized in Figure 5.6. This offers, in contrast to the ABC methods, the opportunity to verify (presuming the posterior distributions are correct), if the underlying parameter distributions are actually Gaussian or if another parametric distribution would be more suitable. To get an intuition on the distribution of V or T, one can, for example, plot a histogram of the MAP estimates. If the marginal posterior densities do not contain much uncertainty, a parametric distribution fitted to the MAP estimates leads already to a reliable estimate of the distribution of V or T. However, if uncertainty is present, a hierarchical method is better suited to capture the overall uncertainty.

Figure 5.7 shows the MCMC and MCMC(MAP) marginal posterior distributions for the noise standard deviations σ^I and σ^ω. For MCMC they overestimate a bit, which corresponds to the fact that the Markov Chain did not converge properly. For

	mean($\hat{R}$)		$\hat{R}$					
	V	T	m_V	σ_V	m_T	σ_T	σ_I	σ_ω
MCMC	3.446	3.515	1.107	1.105	1.185	1.227	2.765	1.684
MCMC(MAP)	1.328	1.319	1.000	1.005	1.005	1.001	1.003	1.002
	mean(n_{eff})		n_{eff}					
	V	T	m_V	σ_V	m_T	σ_T	σ_I	σ_ω
MCMC	2	2	13	12	7	5	2	3
MCMC(MAP)	8	8	2,215	2,114	1,679	2,353	1,297	1,521

Table 5.2.: Gelman-rubin statistics $\hat{R}$ and number of effective samples n_{eff} for MCMC and MCMC(MAP), based on the last 5000 samples of 3 parallel sampled chains, each of total length 11.000, for $N = 100$. Note that the values presented for V and T are $mean(\hat{R}(V)) := \frac{1}{N}\sum_{i=1}^{N}\hat{R}(V_i)$, respectively for T, and analogously for n_{eff}.

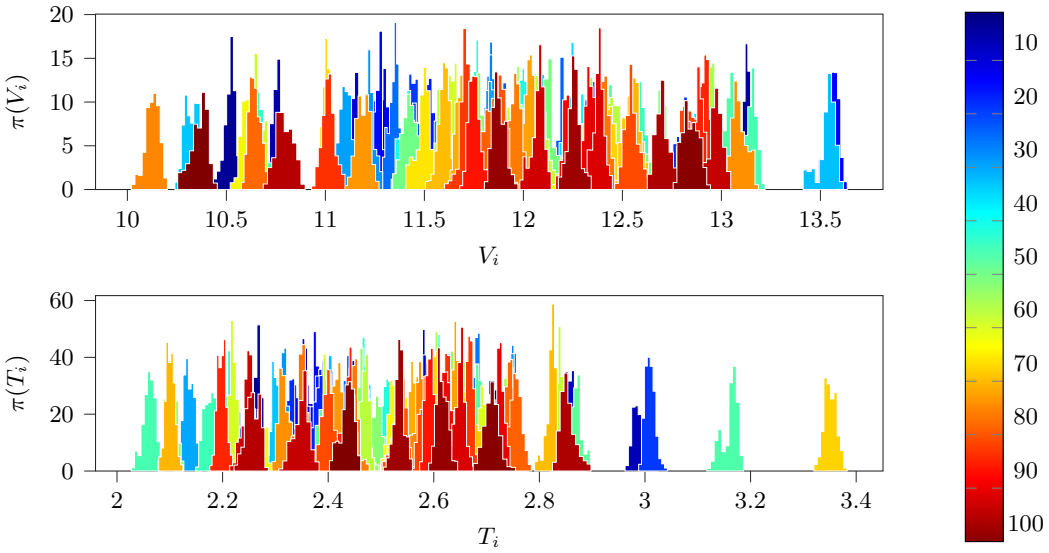


Figure 5.6.: Marginal MCMC(MAP) samples for V_i and T_i for $i = 1, ..., 100$. The color bar indicates the index i.

MCMC(MAP) they are almost identical to the reference values, which is expected in this synthetic case.

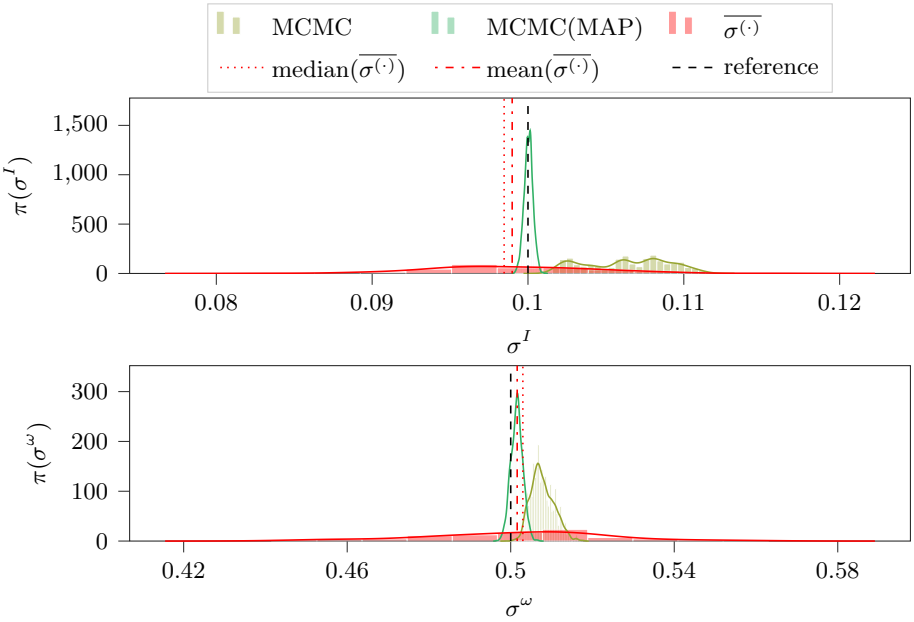


Figure 5.7.: Marginal MCMC and MCMC(MAP) samples for the noise standard deviations σ^I and σ^ω. Also the reference values, a histogram of the estimates $\overline{\sigma^{(\cdot)}} := [\sigma_i^{(\cdot)}]_{i=1,\dots,N}$ via Equation (2.6) and the corresponding mean and median are displayed.

Definition 10. *For an estimator ϑ of a parameter $\Theta \in \mathbb{R}$ define $Bias(\vartheta) = (\hat{\vartheta} - \Theta)^2$, where $\hat{\vartheta}$ is a point estimate of ϑ, e.g. the mean, median or MAP. Further, define the Mean (Median or MAP) Square Error by $MSE(\vartheta) = Bias(\vartheta) + V[\vartheta]$.*

In order to put the considered methods into perspective with respect to accuracy, uncertainty and computational effort, Figure 5.8 shows the summed Bias and MSE of all hyper-parameters versus the number of model evaluations. The Bias is calculated w.r.t. the median of the marginal posterior distribution of $m_V, \sigma_V, m_T, \sigma_T$ and the reference samples empirical moments. The number of model evaluations for ABC are determined by the product of number of proposals M and the number of Sparse Grid points. Here, for $L^{SG} = 2$ and two dimensions, 17 Sparse Grid nodes are used. For MCMC the number of model evaluations is the number of chains times number of samples m times number of observations N. Sampling in one chain is only possible sequentially. Important to note is that the model evaluations do not directly translate to runtime. For ABC and also for SMC ABC in each population they can easily be executed in parallel, which is not

the case for MCMC. This can be already seen in Table 5.1, where SMC ABC is twice as fast as MCMC(MAP) albeit requiring a larger number of model evaluations. Please note that computational times in Table 5.1 strongly depend on the implementation, however they already give an impression on each methods efficiency. As expected the summed MSE for ABC, SMC ABC and MCMC reduce with the number of model evaluations, i.e. corresponding number of samples, where the major part is due to a reduction of the variance, but also due to a reduction in the summed Bias. For MCMC(MAP) initialized already with the MAP estimates no further reduction of the summed MSE can be achieved, however the summed Bias reduces slightly. Overall, w.r.t. the summed Bias, ABC and SMC ABC achieve results that are only a bit worse than MCMC(MAP) and way better than MCMC, in much less time.

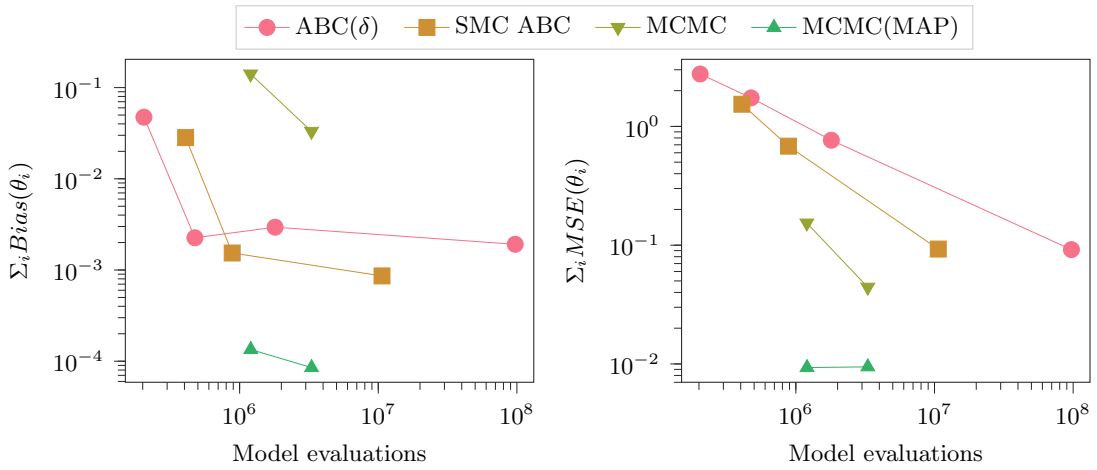


Figure 5.8.: Sum of Bias and MSE, summed over all parameters $\boldsymbol{\theta}$, versus the number of simulation model $\mathcal{M}$ (or metamodel $\mathcal{M}^{PCE}$) evaluations. The Bias is calculated w.r.t. the median of the marginal posterior distribution and the empirical moments of the reference samples. Note that due to parallelization model evaluations do not directly correspond to runtimes.

So far the results presented above were for a fixed data size $N = 100$. In the following we compare the MCMC, MCMC(ABC) and SMC ABC methods for $N = 10, 100, 1000$. Table 5.3 contains measures for the MCMC and MCMC(MAP) sampling efficiency for increasing values of N, but constant length of $11,000$ samples for the three parallel sampled Markov chains. For increasing N the values for $\hat{R}$, n_{eff} and the computational time show that the efficiency decreases drastically, which was already discussed in Section 3.1.1. This is not only due to the high dimensions N but also due to the concentration effect of the posterior distributions as a consequence of highly informative

	MCMC			MCMC(MAP)			SMC ABC
N	$\hat{R}$	n_{eff}	time [s]	$\hat{R}$	n_{eff}	time [s]	time [s]
10	1.13	453	158	1.03	486	151	217
100	3.42	2	400	1.31	62	387	211
1000	5.48	2	3,562	2.23	9	3,536	205

Table 5.3.: Dimension dependency of MH-MCMC sampler efficiency with respect to the number of observations N. Note that the actual number of inferred parameters is $2N + 6$. For concise presentation, the values for $\hat{R}$ and n_{eff} are the mean of all of those parameters statistics. For comparison the runtime of SMC ABC is added.

data. Note that already for the low dimensional case with $N = 10$ the n_{eff} is below 10% of the considered samples. $\hat{R}$ increases in higher dimensional state spaces indicating that the distribution of the Markov chain has not yet converged and the number of samples should be increased. Furthermore, the time needed to execute the algorithm strongly increases with N. We observe a similar behavior for MCMC(MAP), however in a smaller scale. This confirms that the statistical efficiency of the Metropolis-Hastings Algorithm with Gaussian random walk proposal does not perform dimension-independent and justifies the ABC and in particular the SMC ABC approach that improves in performance with increasing N, since the summary statistics improve with increasing N. Also the runtime stays almost constant. For $N = 1000$ SMC ABC is roughly 17 times faster than MCMC(MAP). Figure 5.9 adds on Table 5.3 by presenting the marginal posterior distribution for varying N. For $N = 1000$ the MCMC posterior of m_V and m_T are largely biased and only the SMC ABC and MCMC(MAP) results deliver proper estimates. We also observe that the SMC ABC posterior only slightly concentrates with increasing data. This might be due to the fact that the summary statistics only represent part of the original information. For further uncertainty reduction the summary statistics need to be extended with additional statistics that contain additional information.

Posterior consistency

The proposed method has several layers of approximation, thus we comment in the following on posterior consistency w.r.t. Theorem 3. In order to show posterior consistency, Figure 5.5 presents results of the method for small ABC threshold limit ($\delta \to 0$). Corresponding to the theory the posterior concentrates for decreasing values of δ, which is also summarized in Figure 5.8.

For the considered example the surrogate already has a high approximation quality. Thus, we do not consider posterior consistency with respect to the level L of the

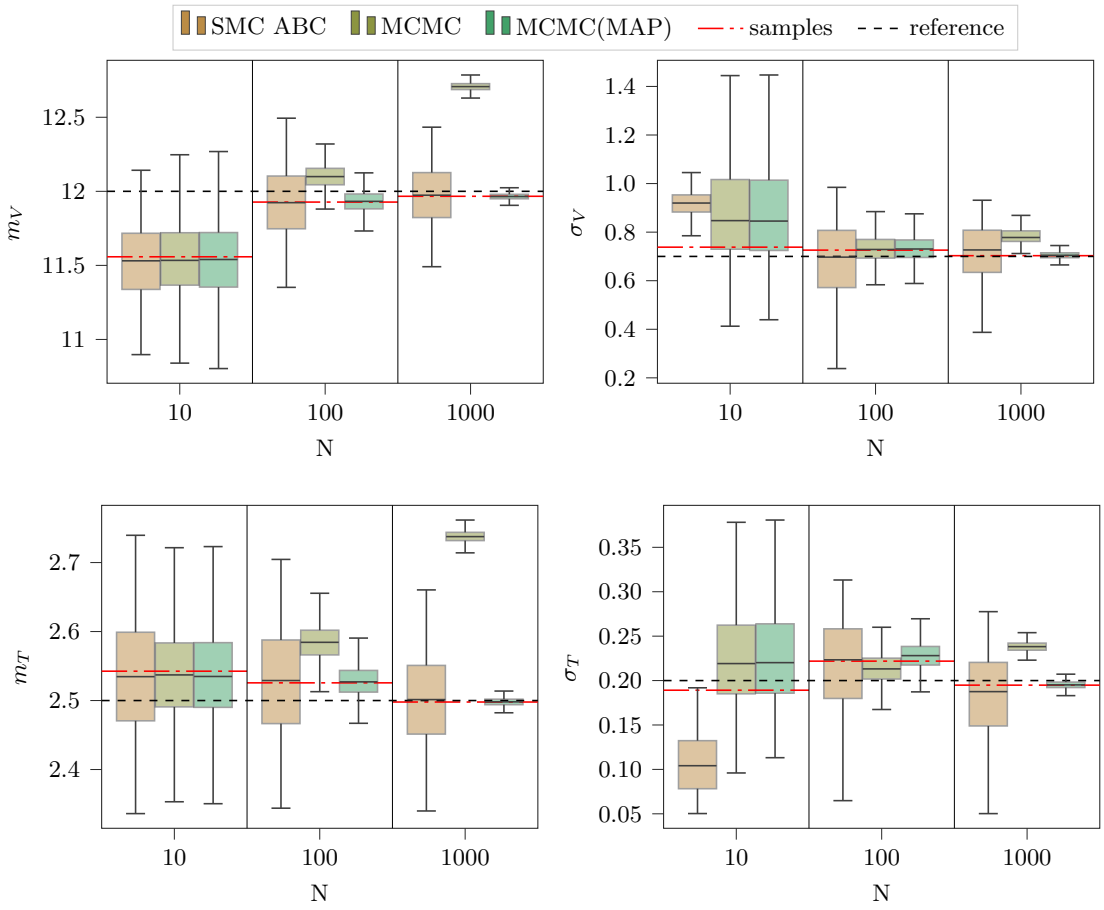


Figure 5.9.: Boxplots of the marginal posterior distributions of m_V, σ_V, m_T, σ_T obtained with SMC ABC, MCMC and MCMC(MAP) for data size $N = 10, 100, 1000$. Also the reference value and sample values depending on N are displayed. Table 5.3 lists the corresponding computational time, $\hat{R}$ and n_{eff}.

surrogate.

Posterior consistency in the small noise limit ($\sigma^I, \sigma^\omega \to 0$) is difficult to show. With decreasing noise in the single measurements also the noise in the summary statistics reduces. Thus the minimal discrepancy value $d(\cdot,\cdot)$ decreases as well. If we now fix the threshold δ for several noise levels the posterior distribution does not concentrate, but rather smear out a bit as more samples are accepted. Consequently, in order to observe posterior consistency in the small noise limit we additionally need to decrease δ. By decreasing δ in the order as the minimal discrepancy value $d(\cdot,\cdot)$ decreases (which is known due to the reference values), we also observe posterior concentration on the empirical moments of the parameter samples.

For the large data limit, i.e. number of observations $N \to \infty$, we observe in Figure 5.9 that the empirical moments of the parameter samples converge to the reference values (due to the law of large numbers) and with this also the posterior distribution shifts more to the reference values. However a concentration can hardly be observed, this might be a consequence of the information loss due to the summary statistics. With increasing N the summary statistics get more and more accurate (again due to the law of large numbers), however they still represent only part of the whole data $\mathcal{Y}$. A second parameter determining the data size is the number of discrete time steps N_t. As it is already very large, $N_t \to \infty$ would not give much insight and is thus not considered in this work.

Increasing the Sparse Grid quadrature level L^{SG} for the summary $S(\boldsymbol{\theta}, \sigma^I, \sigma^\omega; L^{SG})$ theoretically improves approximation quality, but with an increase of computational cost. The presented results are all with $L^{SG} = 2$. Consequently, reducing to $L^{SG} = 1$ would reduce computational costs further. For a detailed analysis of the influence on posterior approximation accuracy and runtime, additional experiments are required in future work.

5.2.3. Test bench data

The test bench was already introduced in detail in Section 2.2. It allows a predefinition of parameter distributions in order to generate measurement data $\mathcal{Y}$. This is used to validate the methods and test them for robustness. In comparison to the artificial setting, inference based on real measurements introduces additional challenges, like model discrepancies and complex structured measurement noise.

An overview of the test bench data $\mathcal{Y}$ is visualized in Figure 2.8. The noise standard deviations σ^I and σ^ω are estimated by taking the median of the estimations $\overline{\sigma_i^{(\cdot)}}$ for each measurement obtained via Equation (2.6) in the stationary time domain $[4, 6]$ seconds. The measurement noise varies for different measurements, which can be observed in Figure 5.11, where a histogram of $\overline{\sigma_i^{(\cdot)}}$ is plotted. Note that for the following analysis, a test bench data set without preprocessing step 1 (outlier removal) is used, which is

visible in the scattering of the noise estimates for σ^{ω}.

PCE surrogate

The simulation model $\mathcal{M}$, based on the complex model (2.12) of the the electric motor test bench, is computationally too expensive to be used for sampling with a reasonable number of samples. Thus a cheaper to evaluate PCE surrogate model is used as introduced in Section 3.3.1. The RMSE scaled by the standard deviation of the validation set for a level $L = 5$ PCE (leads to 181 sparse grid points) is in the range of 10^{-3} for the current and 10^{-4} for the angular velocity. This is sufficient for the following analysis. The speed up by using the surrogate for this example is approximately factor 2440. I.e. evaluation of the original model $\mathcal{M}$ for one sample takes $1.78s \pm 192ms$ (mean $\pm$ std. dev. of 7 runs) and evaluation of the surrogate $\mathcal{M}^{PCE}$ for one sample takes $729\mu s \pm 76.4\mu s$ (mean $\pm$ std. dev. of 7000 runs). Note that depending on the implementation vectorized evaluation of the surrogate is possible. E.g. evaluation of the surrogate $\mathcal{M}^{PCE}$ for 100 samples takes $1.78ms \pm 39.3\mu s$ (mean $\pm$ std. dev. of 700 runs). The speed up compared to the original model is then approximately factor 10^5.

Results

With almost the same methods setting as in the synthetic case Figure 5.10 presents boxplots of the marginal posterior distributions. The only difference is that the thresholds for ABC and SMC ABC are now $\delta \in \{7.5, 5.5, 3.5, 1.5\}$. Table 5.4 displays the runtimes, number of proposals and samples. SMC ABC takes less than half of the time than MCMC(MAP).

	runtime [s]	samples m	proposals M
ABC($\delta = 7.5$)	4	1,500	6,000
ABC($\delta = 5.5$)	9	1,500	13,000
ABC($\delta = 3.5$)	41	1,500	66,000
ABC($\delta = 1.5$)	7,472	1,500	11,968,000
SMC ABC	315	1,500	484,000
MCMC	710	n_{eff}	3*11000
MCMC(MAP)	700	n_{eff}	3*11000

Table 5.4.: Runtimes in seconds, number of samples m and proposals M of the considered methods. For MCMC the number of effective samples n_{eff} is given in detail in Table 5.5.

The results are comparable to the synthetic case, however there is an offset to the reference values for σ_V, m_T, σ_T. This is most likely due to the varying noise structure and model discrepancy. Further, we notice a difference in the SMC ABC and MCMC(MAP)

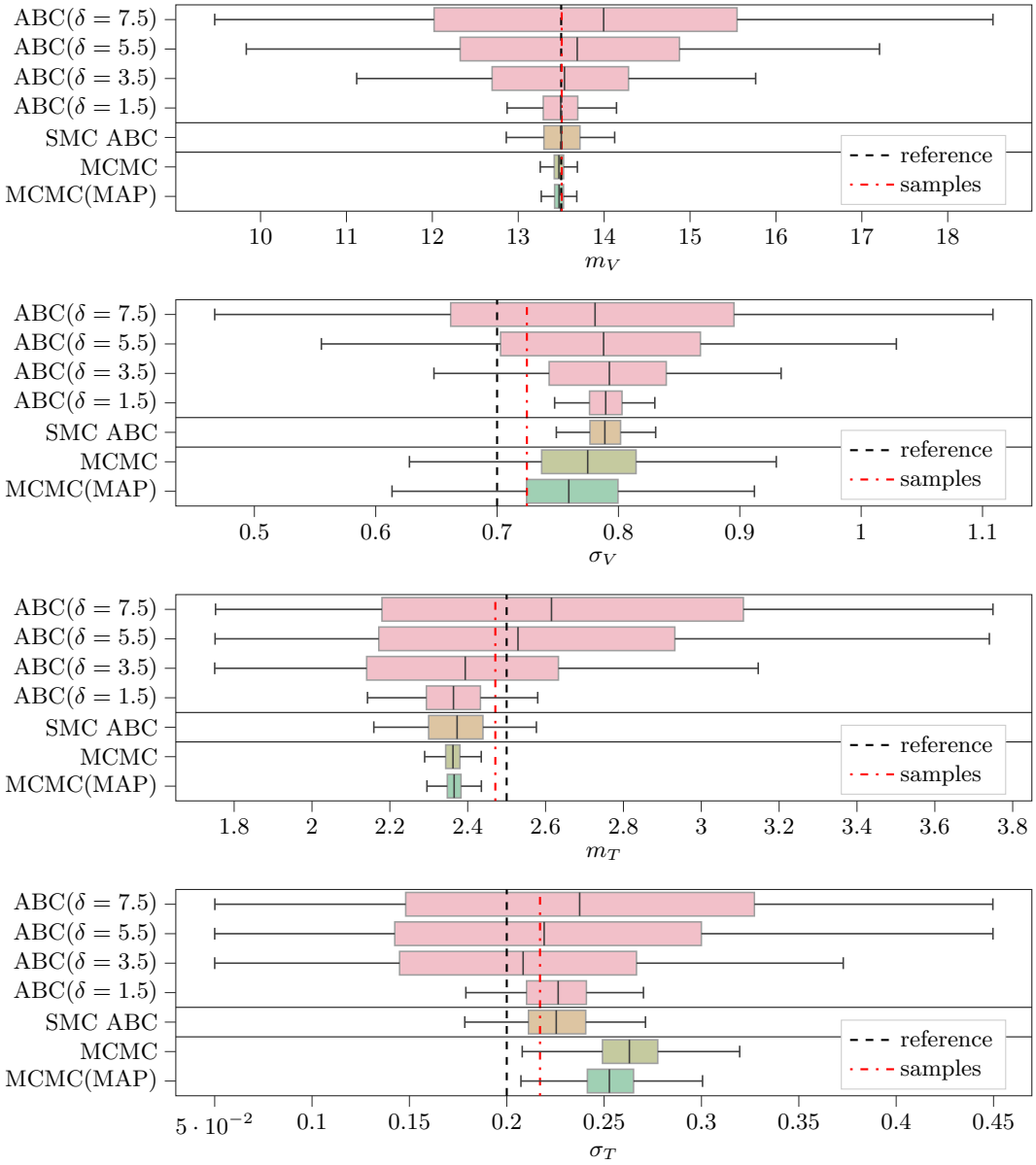


Figure 5.10.: Boxplots of the samples approximating the hyper-parameters marginal posterior distributions in the test bench case obtained with ABC, SMC ABC, MCMC and MCMC(MAP) in comparison to the reference values and sample mean and standard deviation.

posterior distributions for σ_V and σ_T. This is due to the estimation of the noise standard deviations σ_I, σ_ω and due to the additive structure in the summary statistic, see Equation (5.11). Figure 5.11 shows the marginal MCMC and MCMC(MAP) samples for the noise standard deviations σ^I and σ^ω. Note that there is no reference value available. The structure of the noise is not as regular as in the artificial setting and can not be perfectly described by the assumption of independent and identically distributed noise in time and for every measurement. The MCMC estimates have higher values as the median of the estimates $\overline{\sigma_i^{(\cdot)}}$ obtained from the stationary time domain, which are used for ABC and SMC ABC. This leads to the difference in the posterior distributions.

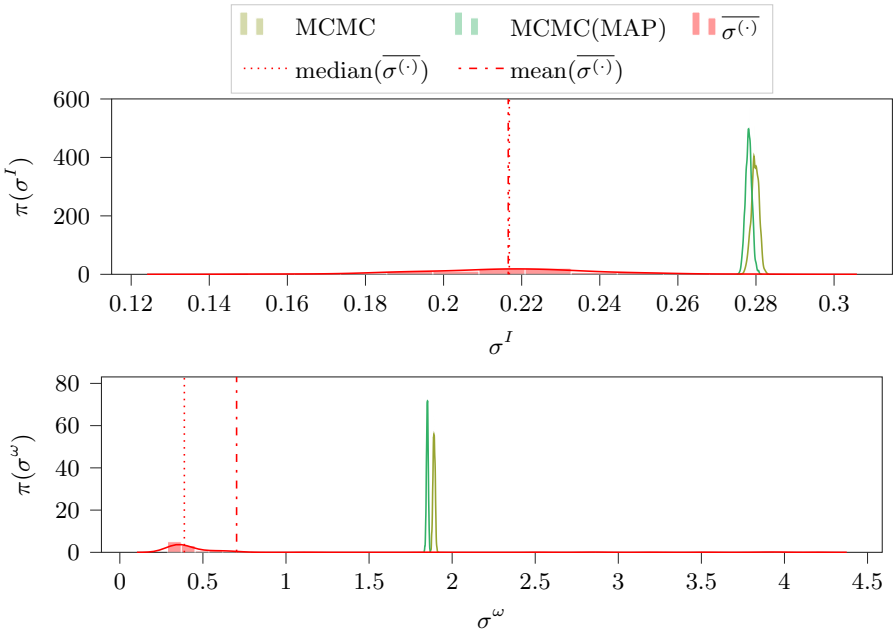


Figure 5.11.: Marginal MCMC and MCMC(MAP) samples for the noise standard deviations σ^I and σ^ω for the test bench. Also a histogram of the estimates $\overline{\sigma^{(\cdot)}} := [\overline{\sigma_i^{(\cdot)}}]_{i=1,\dots,N}$ via Equation (2.6) and the corresponding mean and median are displayed.

Marginal posterior distributions of V_i and T_i for $i = 1, \dots, N$ obtained via MCMC(MAP) are visualized in Figure 5.12. For T_i they are wider as in the synthetic case which is due to larger noise in the data. This slightly simplifies sampling for MCMC and MCMC(MAP), which can also be observed for the sampler efficiency statistics in Table 5.5. Overall we see again a good improvement from MCMC to MCMC(MAP), but even for MCMC(MAP) n_{eff} is still below 15%, which recommends to increase the sample size.

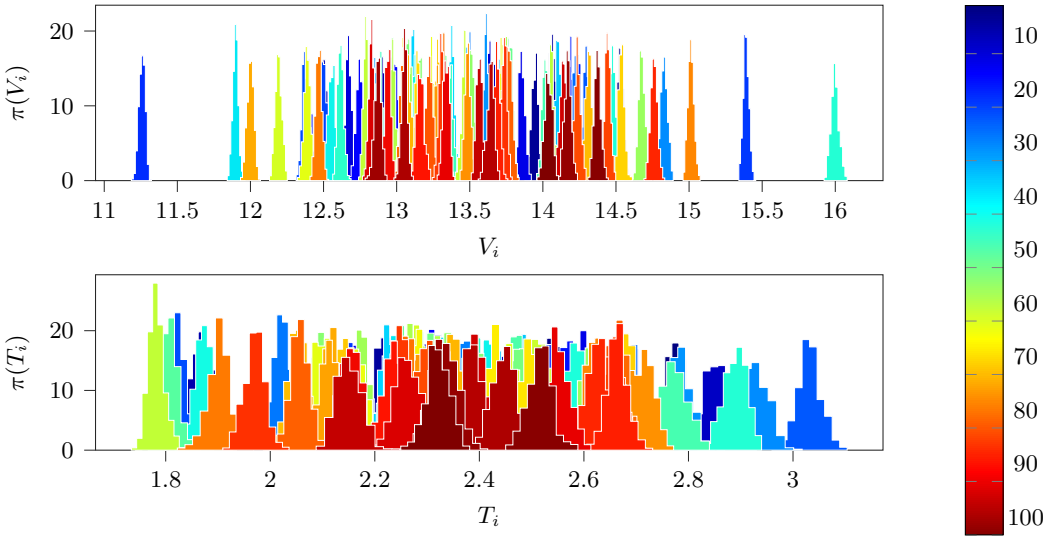


Figure 5.12.: Marginal MCMC(MAP) samples for V_i and T_i for $i = 1, ..., 100$ based on test bench data. The color bar indicates the index i.

	mean($\hat{R}$)		$\hat{R}$					
	V	T	m_V	σ_V	m_T	σ_T	σ_I	σ_ω
MCMC	2.257	1.835	1.003	1.033	1.000	1.169	1.270	1.219
MCMC(MAP)	1.036	1.036	1.000	1.001	1.003	1.002	1.001	1.001
	mean(n_{eff})		n_{eff}					
	V	T	m_V	σ_V	m_T	σ_T	σ_I	σ_ω
MCMC	4	5	2,852	132	1,667	7	5	6
MCMC(MAP)	45	49	2,085	2,138	1,804	1,985	1,297	1,237

Table 5.5.: Gelman-rubin statistics $\hat{R}$ and number of effective samples n_{eff} for MCMC and MCMC(MAP), based on the last 5000 samples of 3 parallel sampled chains, each of total length 11.000, for $N = 100$. Note that the values presented for V and T are $mean(\hat{R}(V)) := \frac{1}{N}\sum_{i=1}^{N}\hat{R}(V_i)$, respectively for T (analog for n_{eff}).

5.2.4. Discussion

Overall, ABC and SMC ABC for the synthetic data achieve results that are comparable to MCMC(MAP) in much less computational time. This advantage increases even more for large data size N. For an increase of data size and consequently increasing number of parameters in our examples the ABC method performs better for almost constant computational cost. This is because the summary statistics and noise standard deviation estimation quality improve, which then improves ABC sampling. For MCMC it is vice versa: performance decreases and also computational costs raise, due to the high dimensions and the concentration effect. The MCMC performance can be improved by providing curvature information and appropriate starting values (with MAP and inverse Hessian initialization), but still the samples are highly autocorrelated.

Of course, plain rejection ABC with small threshold δ is very inefficient, but nevertheless obtains decent results. In order to get a rough overview on posterior distribution, ABC with moderate δ is extremely fast. This fact is further exploited by using a population based SMC ABC method with a sequence of decreasing δ's, where results comparable to plain rejection ABC with small δ with roughly only 10% samples are obtained. As ABC is embarrassingly parallel, further multiprocessing would easily yield additional efficiency gains, whereas potential efficiency gains for MCMC are lower.

The results for the test bench data are comparable to the synthetic case. The offset to the reference values for σ_V, m_T, σ_T is most likely due to the varying noise structure and model discrepancy. In order to improve the MCMC results accuracy one can introduce a more complex noise model, which leads to a higher number of parameters and consequently to even more difficulties in sampling. To improve ABC and SMC ABC one can refine the noise standard deviation estimation by not only approximating them on the stationary time interval $[4, 6]$, but on the whole time interval used for inference. Currently, the noise standard deviation is underestimated, which potentially contributes to an overestimation of σ_V, σ_T, because of the additivity in (5.11). Due to the dynamics of the data in the whole time interval, a more sophisticated approximation method is then required, e.g. kernel ridge regression or GP regression.

The bias in the mean m_T of the load torque T is most likely a consequence of model discrepancy. Chapter 4 already treated model discrepancy in combination with deterministic parameter estimation and Chapter 6 will treat it in combination with aleatoric parameter estimation.

In general, the inference results of the ABC method are strongly influenced by the choice of the summary statistics as well as by the distance function. With multiple input distributions and multiple output quantities of interest scaling of the summary statistics is important. E.g. in [Jung and Marjoram, 2011] they showed that appropriate weighting improves posterior approximation. In our example the outputs are much more sensitive to variations of the voltage than to variations of the mechanical loading. This issue could be addressed in future work with a sensitivity analysis in order to choose an appropriate

distance function. For example, Sobol sensitivity indices [Sobol, 2001, Saltelli et al., 2008] could be used to weight the distance function pro-rata.

In this work the mean and standard deviation are used as summary statistics. In future work, other statistics could be included to the summary statistics in order to increase the information content and further improve the ABC results. However, the selection of summary statistics is non-trivial as already discussed earlier. An optimal selection of summary statistics based on minimum entropy [Nunes and Balding, 2010] was computationally not feasible for this work. Similarly, only the first and second moment of the unknown aleatoric parameters are estimated. Future work might address higher moment approximation to infer a broader class of probability distributions.

Chapter 6.

Aleatoric parameter and model discrepancy estimation

> Knowing is not enough, we must apply. Willing is not enough, we must do.
>
> Bruce Lee (1940-1973)

Adding on aleatoric parameter estimation in the previous Chapter 5, we now explicitly consider model discrepancy in the stochastic inverse problem (see Definition 5). In contrast to Chapter 4, where model discrepancy is inferred for unknown but deterministic simulation model parameters (see inverse problem Definition 4), now model discrepancy needs to be inferred for unknown aleatoric simulation model parameters. Consequently, it does not only depend on time but also on the simulation model parameters and needs to be estimated for the whole support of the aleatoric parameters. Additionally, of course, the challenges from Chapter 4 (identifiability problem, model selection, high computational cost) and Chapter 5 (infinite dimensional problem, high computational costs) are inherited and even amplified.

Related work on model discrepancy correction in stochastic simulators is sparse to non-existent, in particular with joint inference of aleatoric parameters. The majority of work handling model uncertainty in stochastic simulation focuses on covering model discrepancy by input uncertainty, e.g. by assuming and defining input probability distributions, see [Plumlee and Lam, 2017] and the references therein. Otherwise, it is commonly conducted together with model validation during model development. Lam et al. [Lam et al., 2017] detour from this common approach and improve prediction from stochastic simulation via model discrepancy learning in the context of operations research. It builds on and generalizes previous work [Plumlee and Lam, 2016, Plumlee and Lam, 2017] (both for finite-dimensional, discrete system responses). They utilize the idea of inferring model discrepancies in deterministic computer experiments to improve the prediction of stochastic simulation. With the assumption that system observations at design points $\boldsymbol{x} \in \mathcal{X}$ are available, they smartly combine regression on a collection of summary statistics to infer the model discrepancy distribution.

We follow the basic idea of this approach, but loosen the assumption on knowledge of design points. The data considered in this thesis has unknown design points, i.e. unknown

model parameters $\boldsymbol{x} \in \mathcal{X}$. Section 6.1 introduces hierarchical aleatoric parameter and model discrepancy estimation as an efficient combination of the methods presented in the previous chapters. In order to reduce computational cost we propose a method to build a model discrepancy surrogate on a representative subset of the data only. Finally, numerical results for the proposed hierarchical surrogate-based discrepancy-corrected inference method are presented for the test bench in Section 6.2.

Due to the combination of the already detailed Chapters 3, 4 and 5, this chapter concentrates only on the necessary methods for efficient combination and else refers to the previous parts of this thesis for brevity.

6.1. Efficient combination of aleatoric parameter and model discrepancy estimation

In Section 5.1 we presented methods to infer the simulation model parameter distribution $\pi(X)$ from $\mathcal{Y}$ in a hierarchical surrogate-based way, either with MCMC or with ABC and summary statistics, but without consideration of a model discrepancy term. In Section 4.3 we presented methods to infer for one measurement signal $\boldsymbol{y} \in \mathcal{Y}$ the simulation model parameters $\boldsymbol{x}$ and the model discrepancy term $\boldsymbol{\delta}$. Now we combine these two approaches to infer the simulation model parameter distribution $\pi(X)$ and the model discrepancy term $\boldsymbol{\delta}(X)$ from $\mathcal{Y}$. We explicitly consider model discrepancy in the stochastic inverse problem (see Definition 5) by

$$\Upsilon \overset{d}{=} \mathcal{M}(X) + \boldsymbol{\delta}(X) \tag{6.1}$$

where the distributions of X and $\boldsymbol{\delta}(X)$ are unknown. W.r.t. the available noisy observations $\boldsymbol{y}_i \in \mathcal{Y}$ of the true underlying process Υ, it can be reformulated to

$$\boldsymbol{y}_i = \mathcal{M}(\boldsymbol{x}_i) + \boldsymbol{\delta}_i(\boldsymbol{x}_i) + \boldsymbol{\varepsilon}_i \tag{6.2}$$

for $i = 1, \dots, N$, with unknown $\boldsymbol{x}_i, \boldsymbol{\delta}_i(\boldsymbol{x}_i)$ and $\boldsymbol{\varepsilon}_i$ (similar to (4.8)). Note that $\boldsymbol{x}_i = X(\omega)$ are realizations of the random variable X. Also, $\mathcal{M}(\boldsymbol{x}_i)$ and $\boldsymbol{\delta}_i(\boldsymbol{x}_i)$ are realizations of $\mathcal{M}(X)$ and $\boldsymbol{\delta}(X)$, such that they explain the data $\boldsymbol{y}_i$ together with the noise term $\boldsymbol{\varepsilon}_i$. Remember that $\boldsymbol{\delta}_i(\boldsymbol{x}_i) = \boldsymbol{\delta}_i(\boldsymbol{x}_i, \Delta)$ is already discretized in time and the dependence on the mesh Δ is omitted for notational convenience.

Joint inference of a model discrepancy $\boldsymbol{\delta}(X)$ for all $\boldsymbol{y}_i \in \mathcal{Y}$ is very complex and computationally unfeasible. We suggest to solve (6.2) independently for some $\boldsymbol{y}_i \in \mathcal{Y}$. The subscript i shall explicitly account the dependence of $\boldsymbol{\delta}_i$ on $\boldsymbol{x}_i$, since not the full model discrepancy $\boldsymbol{\delta}(\boldsymbol{x})$ for all $\boldsymbol{x} \in \mathcal{X}$ is inferred, but only for specific $\boldsymbol{x}_i$'s.

Inference of both, $\boldsymbol{x}_i$ and $\boldsymbol{\delta}_i(\boldsymbol{x}_i)$, for all $\boldsymbol{y}_i \in \mathcal{Y}$ is still burdened with high computational costs. Thus, we suggest an approach with reduced cost, where $\boldsymbol{x}_i$ and $\boldsymbol{\delta}_i(\boldsymbol{x}_i)$ are inferred for a representative subset of $\mathcal{Y}$ only. This is then used to build a model discrepancy surrogate in order to approximate for the remaining measurements.

6.1.1. Divergence-based subset selection

Subset selection, also denoted by feature selection or dimension reduction, is an often occurring core issue in many problems, such as regression, supervised learning or pattern recognition. It is concerned with the task of identifying a subset of data points (features) that are relevant for the task at hand. The learning method then focuses on the subset and ignores the rest. Focusing on relevance alone is insufficient for efficient feature selection of high-dimensional data and should be complemented by redundancy analysis, e.g. with correlation-based methods, see [Yu and Liu, 2004]. For details on subset selection of regression variables, see [Miller, 2002]. And for further introduction and categorization of feature selection methods in Machine Learning context, we refer to [Guyon and Elisseeff, 2003, Zhang et al., 2013]. The latter work proposes divergence-based feature selection methods for classification, which improve efficiency and reduce cost. We roughly follow the divergence-based approach and introduce our subset selection approach in the following.

Remember that $\mathcal{Y}$ consists of $\boldsymbol{y}_i,\ i = 1, \dots, N$ with $\boldsymbol{y}_i = [\boldsymbol{y}_i^I, \boldsymbol{y}_i^\omega]$. For a subset of indices $\mathcal{I} \subset \{1, \dots, N\}$ define by

$$\mathcal{Y}_\mathcal{I} := \{\boldsymbol{y}_i \in \mathcal{Y} : i \in \mathcal{I}\} \tag{6.3}$$

a subset of measurement signals. Further let $\pi_\mathcal{Y}$ denote the probability distribution of the data $\mathcal{Y}$. For the selection of a representative data subset $\mathcal{Y}_\mathcal{I}$, we need to find a subset of indices $\mathcal{I}$ with $N_\mathcal{I} = |\mathcal{I}|$ minimal, such that $\pi_{\mathcal{Y}_\mathcal{I}}$ approximates $\pi_\mathcal{Y}$ well w.r.t. some objective.

Inference of $\boldsymbol{x}_i$ and $\boldsymbol{\delta}_i(\boldsymbol{x}_i)$ for each $\boldsymbol{y}_i \in \mathcal{Y}_\mathcal{I}$ should lead to a set $\{\boldsymbol{x}_i\}_{i\in\mathcal{I}}$ that covers the model parameter space $\mathcal{X}$ well. To achieve this, $\mathcal{I}$ should be such that $\mathcal{Y}_\mathcal{I}$ covers best the full output space of $\boldsymbol{y}_i^I$ and $\boldsymbol{y}_i^\omega$ in $\mathcal{Y}$.

A first objective function for an optimal subset of indices $\mathcal{I}^{opt}$ is

$$\mathcal{I}^{opt} = \arg \min_{\mathcal{I}\subset\{1,\dots,N\}} \mathrm{d}_{\mathrm{KL}}\left(\pi_\mathcal{Y} \| \pi_{\mathcal{Y}_\mathcal{I}}\right) + \lambda|\mathcal{I}|, \tag{6.4}$$

for a $\lambda > 0$. $\mathrm{d}_{\mathrm{KL}}(\cdot\|\cdot)$ is the Kullback-Leibler (KL) divergence. Note that d_{KL} is not symmetric and that we choose on purpose this kind of order of the arguments, as this enforces that the approximation $\pi_{\mathcal{Y}_\mathcal{I}}$ is non-zero wherever $\pi_\mathcal{Y}$ is non-zero. Consequently the support of $\pi_\mathcal{Y}$ should be preserved with $\pi_{\mathcal{Y}_\mathcal{I}}$. In contrast, the reversed order of arguments in d_{KL} would yield an approximation $\pi_{\mathcal{Y}_\mathcal{I}}$ of one of the modes of $\pi_\mathcal{Y}$, but eventually with an underestimation of the support. For further details on the Kullback-Leibler divergence we refer to [Bishop, 2006] and for an illustration of this property in particular to Ch. 10, Figure 10.3.

The objective in (6.4) leads to a $\pi_{\mathcal{Y}_\mathcal{I}}$ that not only preserves the support of $\pi_\mathcal{Y}$, but also the shape. However, selected subset samples are clustering in high probability regions and in low probability regions there are only a few samples. This might yield a bad

representation of the parameter space, in particular in higher dimensions. To obtain a $\mathcal{Y}_\mathcal{I}$ that is more uniformly distributed, we modify the objective. For an $\alpha > 0$ denote by

$$\pi_\mathcal{Y}^{trunc}(\boldsymbol{y}) = \begin{cases} \alpha, & \text{if } \pi_\mathcal{Y}(\boldsymbol{y}) \geq \alpha, \\ \pi_\mathcal{Y}(\boldsymbol{y}), & \text{else,} \end{cases} \tag{6.5}$$

the truncated probability distribution of the data (with additional normalization to be a proper probability distribution). Then, the second objective

$$\mathcal{I}^{opt} = \arg\min_{\mathcal{I}\subset\{1,\dots,N\}} \mathrm{d_{KL}}\left(\pi_\mathcal{Y}^{trunc} \| \pi_{\mathcal{Y}_\mathcal{I}}\right) + \lambda|\mathcal{I}|, \tag{6.6}$$

leads also to a support preserving, but more uniform subset. The uniformity depends on α. Later, Example 4 illustrates the difference of those two objective functions in Figure 6.1.

The optimization problem is complex to solve. An exhaustive search over all $2^N - 1$ candidate subsets is computationally intractable. The approach in this work is to fix $N_\mathcal{I}$, draw m samples $\mathcal{I}_j \subset \{1,\dots,N\},\ j = 1,\dots,m$ and select the $\mathcal{I}_j$ that minimizes the objective. As $\pi_\mathcal{Y}$ and $\pi_{\mathcal{Y}_\mathcal{I}}$ are not available, kernel density estimates (KDE) based on $\mathcal{Y}$ and $\mathcal{Y}_\mathcal{I}$ are used. The KDE is equipped with a Gaussian kernel and Scott's heuristic for automatic bandwidth determination, see [Scott, 2015] for details.

To reduce computational costs further we suggest following heuristic: For each $\boldsymbol{y}_i = [\boldsymbol{y}_i^I, \boldsymbol{y}_i^\omega] \in \mathcal{Y}$ calculate the mean of $\boldsymbol{y}_i^{(\cdot)}$ in the stationary time interval $[4, 6]$ seconds according to Definition 1. This yields a set of N two dimensional points

$$\overline{\mathcal{Y}} := \{(\overline{\boldsymbol{y}_i^I}, \overline{\boldsymbol{y}_i^\omega})\}_{i=1,\dots,N} \in \mathbb{R}^{2\times N}. \tag{6.7}$$

$\overline{\mathcal{Y}}$ is used to compute a KDE $\pi_{\overline{\mathcal{Y}}}^{KDE}$ of the underlying probability distribution $\pi_{\overline{\mathcal{Y}}}$, where the latter can be seen as an approximation of $\pi_\mathcal{Y}$ in a stationary time interval. Objective (6.6) with $\pi_{\overline{\mathcal{Y}}}^{KDE,trunc}$ is then used to determine a subset $\mathcal{I}$ via sampling.

Yet another heuristic improves sampling. Sampling $\mathcal{I}_j \subset \{1,\dots,N\}$ with probability proportional to $1/\pi_{\overline{\mathcal{Y}}}^{KDE}(\boldsymbol{y}_i)$ for each $i \in \{1,\dots,N\}$ yields more often samples with small objective, compared to uniform sampling from $\{1,\dots,N\}$. This is illustrated in Figure 6.2 with the setting of Example 4.

Example 4 (Subset selection 1D example)**.** Let $\mathcal{Y}$ be a set of $N = 100$ samples $y_i \in \mathbb{R}$ from a Gaussian $\mathcal{N}(0, 1)$. Figure 6.1 displays the samples of $\mathcal{Y}$ with its KDE $\pi_\mathcal{Y}^{KDE}$ and marks the samples of the selected subsets $\mathcal{Y}_\mathcal{I}$ with $N_\mathcal{I} = 10$. $\mathcal{Y}_\mathcal{I}$ is selected with objective (6.4) (left) and objective (6.6) with $\alpha = 0.1$ (right). The best out of $m = 1000$ samples of $\mathcal{I}_j$, uniformly from $\{1,\dots,N\}$, are used. Both subsets preserve the support, but the one obtained with objective (6.6) is, as desired, more uniform. Figure 6.2 compares the objective (6.6) of the samples $\mathcal{I}_j$ via uniform sampling and via improved sampling according to the inverse KDE of the data for subset size $N_\mathcal{I} = 10$ (left) and $N_\mathcal{I} = 20$ (right).

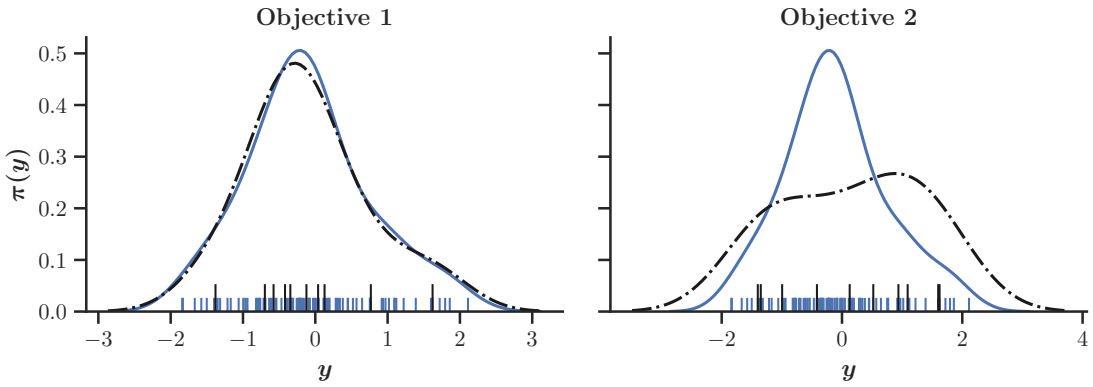


Figure 6.1.: Comparison of subset selection with objective (6.4) (left) and (6.6) (right). Both plots show the data $\mathcal{Y}$ from Example 4 with its KDE (blue) and the subset $\mathcal{Y}_{\mathcal{I}}$ with its KDE (black, dash-dotted). Both preserve the support. The left is also nicely preserving the shape, but with most samples in high probability region, whereas the right yields more uniformly distributed samples.

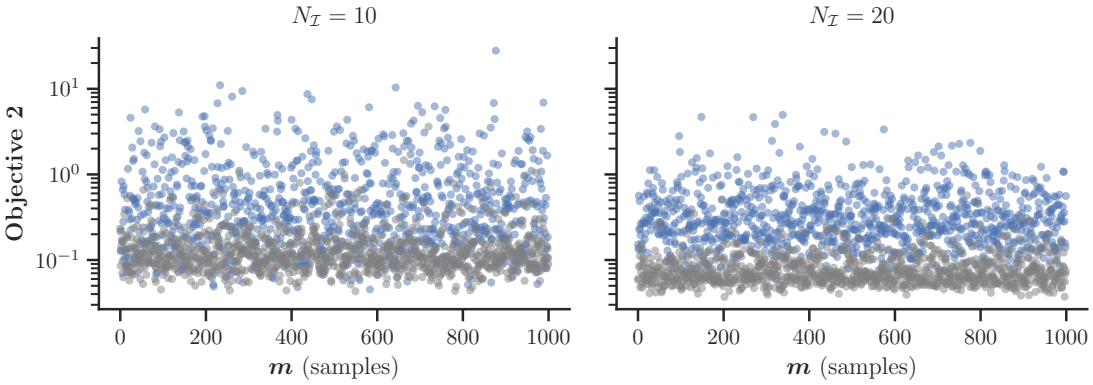


Figure 6.2.: Comparison of uniform sampling (blue) and sampling according to the inverse KDE of the data (gray) for subset selection with objective (6.6) for subset size $N_{\mathcal{I}} = 10$ (left) and $N_{\mathcal{I}} = 20$ (right).

6.1.2. Model discrepancy surrogate

With a representative subset $\mathcal{Y}_\mathcal{I}$ of $\mathcal{Y}$, inference of the simulation model parameters $\boldsymbol{x}_i$, the model discrepancies $\boldsymbol{\delta}_i(\boldsymbol{x}_i)$ and noise term $\boldsymbol{\varepsilon}_i$ need to be carried out for each measurement $\boldsymbol{y}_i \in \mathcal{Y}_\mathcal{I}$. This involves for each $i \in \mathcal{I}$ the challenging identifiability problem and requires careful solution strategies.

The method BM2, introduced in Section 4.3.3, was successfully employed for test bench data $\boldsymbol{y}_j$ with $j = 15, 16, 26$ (see Section 4.5). There, the BM2 model discrepancy term $\boldsymbol{\delta}^K(\boldsymbol{a})$ was modeled by weighted Laguerre polynomials with coefficients $\boldsymbol{a}$ and truncation parameter K. BM2 together with Algorithm 5 iteratively leads to approximations of the posterior distributions

$$\pi^{K_i}(\boldsymbol{x}_i, \boldsymbol{a}_i, \sigma_i^I, \sigma_i^\omega | \boldsymbol{y}_i), \quad \forall i \in \mathcal{I}. \tag{6.8}$$

Let $\hat{\boldsymbol{x}}_i$ and $\hat{\boldsymbol{\delta}}_i(\hat{\boldsymbol{x}}_i) := \boldsymbol{\delta}^{K_i}(\hat{\boldsymbol{a}})$ denote the posterior mean. The set of model parameter-discrepancy pairs

$$\mathcal{D} := \{(\hat{\boldsymbol{x}}_i, \hat{\boldsymbol{\delta}}_i(\hat{\boldsymbol{x}}_i))\}_{i \in \mathcal{I}}, \tag{6.9}$$

can then be used to build a surrogate $\hat{\boldsymbol{\delta}}(\boldsymbol{x})$ of the model discrepancy $\boldsymbol{\delta}(\boldsymbol{x})$ for all $\boldsymbol{x} \in \mathcal{X}$. As $\boldsymbol{x} = X(\omega)$, this serves also as surrogate in the stochastic case. In particular, we fit a PCE to $\mathcal{D}$ and compute the PCE coefficients with least squares, see Section 3.3 for details. For PCE construction it is assumed that $\hat{\boldsymbol{x}}_i$ are uniformly sampled from $\mathcal{X}$, similar to the construction of the PCE for $\mathcal{M}$.

Of course, the quality of the model discrepancy surrogate $\hat{\boldsymbol{\delta}}(\boldsymbol{x})$ depends on the representative quality of the subset $\mathcal{Y}_\mathcal{I}$ and of the approximation quality of the set of posterior mean model parameter-discrepancy pairs $\mathcal{D}$.

In some applications it might be possible to obtain reference model parameters $\boldsymbol{x}_i^\dagger$ for some measurements $\boldsymbol{y}_i$ with $i \in \mathcal{I}^\dagger$. Then $\boldsymbol{\delta}_i^K(\boldsymbol{a}_i)$ can be fitted directly to the residuals $\boldsymbol{r}_i = \boldsymbol{y}_i - \mathcal{M}(\boldsymbol{x}_i^\dagger)$ via

$$(\boldsymbol{a}_i^\dagger, K_i^\dagger) = \arg\min_{(\boldsymbol{a}_i, K_i)} \|\boldsymbol{r}_i - \boldsymbol{\delta}_i^K(\boldsymbol{a}_i)\|_{L^2}, \quad i \in \mathcal{I}^\dagger. \tag{6.10}$$

Denote the reference model discrepancy by $\boldsymbol{\delta}_i^\dagger(\boldsymbol{x}_i^\dagger) := \boldsymbol{\delta}_i^{K_i^\dagger}(\boldsymbol{a}_i^\dagger)$, the reference model parameter-discrepancy set by $\mathcal{D}^\dagger$ and the model discrepancy surrogate by $\boldsymbol{\delta}^\dagger(\boldsymbol{x}), \boldsymbol{x} \in \mathcal{X}$. The extrapolation quality of $\boldsymbol{\delta}^\dagger(\boldsymbol{x})$ depends on the representative quality of the subset $\mathcal{Y}_{\mathcal{I}^\dagger}$ and with this on the number and locations of $\boldsymbol{x}_i^\dagger \in \mathcal{X}, i \in \mathcal{I}^\dagger$.

It might be that only a small number of reference model parameters are available or that they are not sufficient to cover the model parameter space $\mathcal{X}$. Then a combination of $\mathcal{D}$ and $\mathcal{D}^\dagger$ is advisable. $\mathcal{I}^\dagger$ can be used as a starter for subset selection to obtain a joint subset $\mathcal{Y}_\mathcal{J}$ with $\mathcal{J} = \mathcal{I}^\dagger \cup \mathcal{I}$ that improves the model discrepancy surrogate.

Remark. If available, $\boldsymbol{\delta}^\dagger(\boldsymbol{x})$ might serve, in some sense, as prior for inference of $\{\boldsymbol{x}_i, \boldsymbol{\delta}_i(\boldsymbol{x}_i)\}_{i\in\mathcal{I}}$. But this is non-trivial as $\boldsymbol{x}_i$ is unknown and the approximation quality of $\boldsymbol{\delta}^\dagger(\boldsymbol{x}_i)$ might be bad. This combination requires further research and is out of scope for this work. An idea for further development is an iterative three-stage inference: (1) Assume $\boldsymbol{\delta}^\dagger(\boldsymbol{x})$ is correct and infer $\boldsymbol{x}_i^{(1)}$ from $\boldsymbol{y}_i = \mathcal{M}(\boldsymbol{x}_i) + \boldsymbol{\delta}^\dagger(\boldsymbol{x}_i) + \boldsymbol{\varepsilon}_i$. (2) Use $\boldsymbol{\delta}^\dagger(\boldsymbol{x}_i^{(1)})$ as prior for $\boldsymbol{\delta}^{K_i}(\boldsymbol{a}_i)$ and infer $\boldsymbol{x}_i^{(2)}$ from $\boldsymbol{y}_i = \mathcal{M}(\boldsymbol{x}_i) + \boldsymbol{\delta}^{K_i}(\boldsymbol{a}_i) + \boldsymbol{\varepsilon}_i$. (3) Use the posterior mean of $(\boldsymbol{x}_i^{(2)}, \boldsymbol{\delta}^{K_i}(\boldsymbol{a}_i))$ to correct $\boldsymbol{\delta}^\dagger(\boldsymbol{x})$. Repeat for $i = i + 1$.

6.1.3. Hierarchical surrogate-based discrepancy-corrected inference

With the model discrepancy surrogate $\hat{\boldsymbol{\delta}}(\boldsymbol{x})$ and the simulation model surrogate $\mathcal{M}(\boldsymbol{x})$ we basically perform hierarchical surrogate-based inference as in Section 5.1. The surrogate-based discrepancy-corrected simulation model

$$\mathcal{M}^{\boldsymbol{\delta}}(\boldsymbol{x}) := \mathcal{M}(\boldsymbol{x}) + \hat{\boldsymbol{\delta}}(\boldsymbol{x}) \tag{6.11}$$

with $\hat{\boldsymbol{\delta}}$ fixed, requires no additional parameters for hierarchical inference and can be used seamlessly with the methods introduced earlier. Figure 6.3 gives an overview on the required steps to perform hierarchical surrogate-based discrepancy-corrected inference.

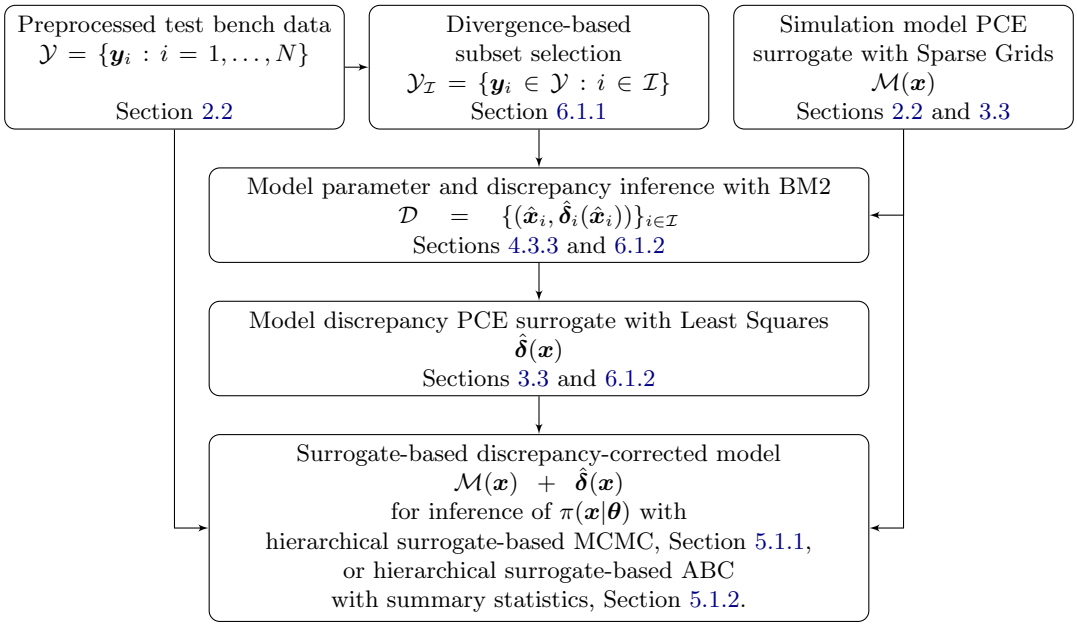


Figure 6.3.: Overview of the required steps to perform hierarchical surrogate-based discrepancy-corrected inference.

If the approximation quality of $\hat{\boldsymbol{\delta}}(\boldsymbol{x})$ is sufficient, then the hierarchical inference leads to corrected parameters and hyper-parameters. Further, $\hat{\boldsymbol{\delta}}(\boldsymbol{x})$ might serve as tool for modelers to refine the simulation model $\mathcal{M}$ in areas where $\hat{\boldsymbol{\delta}}(\boldsymbol{x})$ is large.

6.2. Numerical experiments

The hierarchical surrogate-based discrepancy-corrected inference is applied to the test bench data with corresponding basic model (2.11) (see Section 2.2). In the following Section 6.2.1 details on the inference setup and Section 6.2.2 presents the numerical results.

We follow the abbreviations of the methods introduced in Chapter 4 and Chapter 5. I.e. BM2 denotes the method introduced in Section 4.3.3, MCMC(MAP) the method in Section 5.1.1 and SMC ABC the method in Section 5.1.2.

6.2.1. Inference setup

For inference of model parameters $\boldsymbol{x}_i = (V_i, T_i)$ and model discrepancy $\boldsymbol{\delta}_i(\boldsymbol{x}_i)$ for $i \in \mathcal{I}$ with BM2 the inference setup described in Section 4.5.1 is used. For hierarchical inference of hyper-parameters $m_V, \sigma_V, m_T, \sigma_T$ and all parameters $\boldsymbol{x}_i = (V_i, T_i),\ i = 1, \dots, N$ the inference setup described in Section 5.2.1 is used.

6.2.2. Test bench data

The divergence-based subset selection with objective (6.6) and $N_{\mathcal{I}} = 10$ leads to the data subset $\mathcal{Y}_{\mathcal{I}}$, containing 10% of the data. The selected $\boldsymbol{y}_i, i \in \mathcal{I}$ are displayed in the $\overline{\mathcal{Y}} := \{(\overline{\boldsymbol{y}_i^I}, \overline{\boldsymbol{y}_i^\omega})\}_{i=1,\dots,N}$-plane in Figure 6.4. As the reference values $\boldsymbol{x}_i^\dagger = (V_i^\dagger, T_i^\dagger)$ are known (due to the test bench data generation process), the corresponding $\boldsymbol{x}_i^\dagger = (V_i^\dagger, T_i^\dagger)$ with $i \in \mathcal{I}$ are marked as well. Both, the output and input space are reasonably covered.

The BM2 results for $\boldsymbol{y}_i \in \mathcal{Y}_{\mathcal{I}}$ lead to the set $\mathcal{D}$ and then to the model discrepancy surrogate $\hat{\boldsymbol{\delta}}(\boldsymbol{x})$. Selection of optimal $K_i, i \in \mathcal{I}$ with Algorithm 5 worked reasonably well under consideration of a GP noise standard deviation estimation as stopping criterion, as described in Section 4.5.4. The K_i range from 4 to 10. Note that in some cases, compared to the reference model parameters, other values than the selected yield better results.

For comparison $\boldsymbol{\delta}^\dagger$ is computed as well on the same subset and with $K_i = 7,\ \forall i \in \mathcal{I}$. Hierarchical inference is now carried out with

- simulation model surrogate $\mathcal{M}$ only,
- simulation model plus reference model discrepancy surrogate $\mathcal{M} + \boldsymbol{\delta}^\dagger$

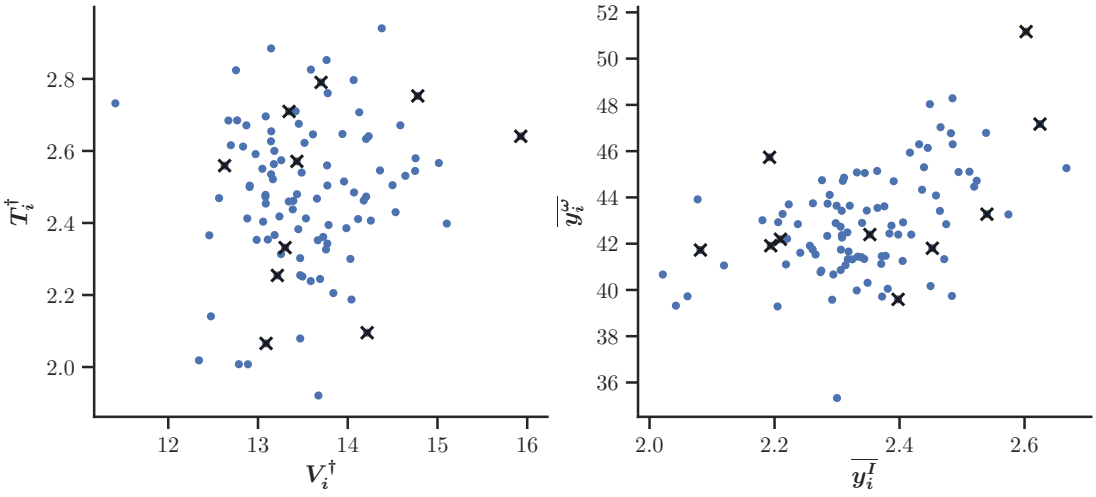


Figure 6.4.: Divergence-based subset selection with objective (6.6) for the test bench data based on $\overline{\mathcal{Y}} := \{(\overline{\boldsymbol{y}_i^I}, \overline{\boldsymbol{y}_i^\omega})\}_{i=1,\dots,N} \in \mathbb{R}^{2\times N}$ (right). Representative for $\boldsymbol{y}_i \in \mathcal{Y}_\mathcal{I}$ the selected data points in $\overline{\mathcal{Y}}$ with $i \in \mathcal{I}$ are marked by $\times$. The corresponding reference values (left) are marked as well by $\times$.

- and simulation model plus inferred model discrepancy surrogate $\mathcal{M} + \hat{\boldsymbol{\delta}}$.

Figure 6.5 displays the MCMC(MAP) results for V_i and T_i, $i = 1, \dots, N$. Inference of V_i corresponds well to the reference values $V_i^\dagger$ for all three models. The inference results for T_i with only $\mathcal{M}$ do not correspond to the reference and are further spread. As expected, this is corrected with $\mathcal{M} + \boldsymbol{\delta}^\dagger$ and also with $\mathcal{M} + \hat{\boldsymbol{\delta}}$. Whereas the latter two results for T_i are slightly different. Overall, the uncertainty in the T_i estimates is higher as for V_i.

Figure 6.6 presents the hierarchical inference results for the hyper-parameters obtained with MCMC(MAP) and SMC ABC. The threshold schedule for SMC ABC is chosen such that a similar amount of proposals ($M \approx 200.000$) are drawn for each of the three models. The SMC ABC runtimes are around 110 seconds and the MCMC(MAP) runtimes around 800 seconds. Analog to the single results for V_i and T_i above, the corrected simulation models lead to a slight improvement in m_T and large improvement in σ_T, where SMC ABC and MCMC(MAP) perform similar. For m_V a slight drift away from the reference is observable for both corrected models.

6.2.3. Discussion

Overall, the hierarchical surrogate-based discrepancy-corrected inference works well for the test bench data and the results with simulation model plus inferred model discrepancy surrogate $\mathcal{M} + \hat{\boldsymbol{\delta}}$ are comparable to those with simulation model plus reference model

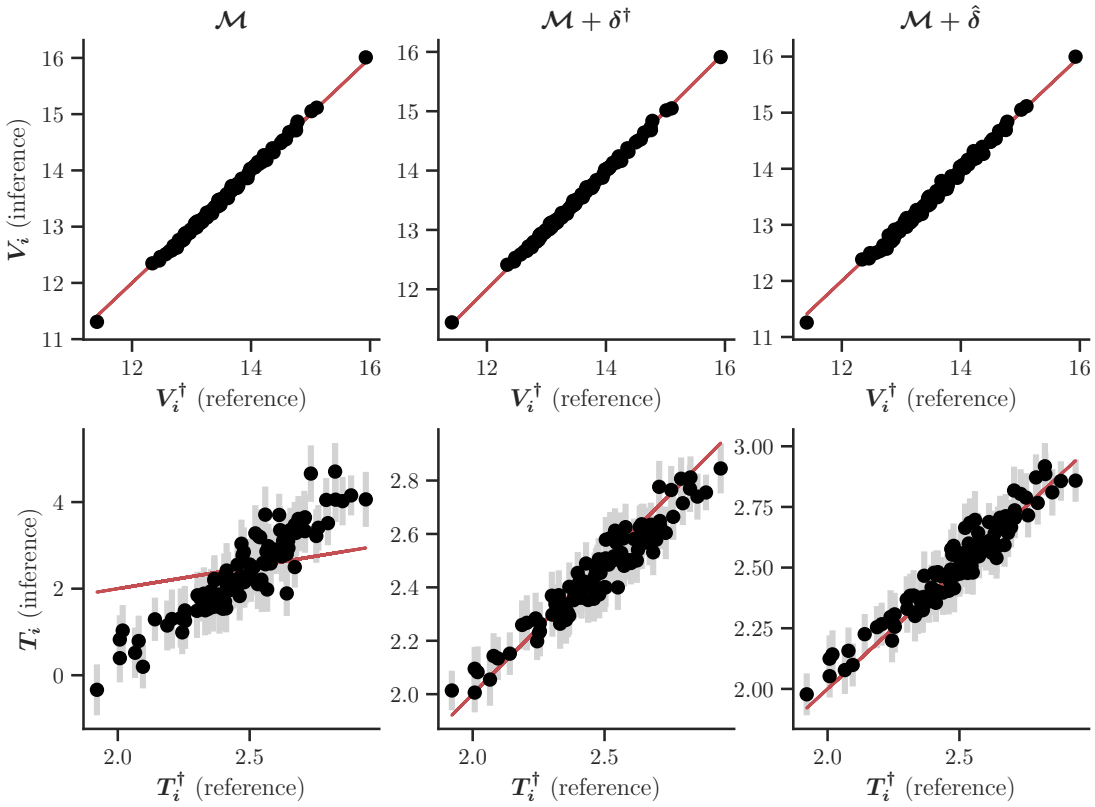


Figure 6.5.: Hierarchical surrogate-based discrepancy-corrected inference for the test bench data. Comparison of MCMC(MAP) results for (top row) V_i and (bottom row) T_i, $i = 1, \ldots, N$ with (left column) only simulation model surrogate $\mathcal{M}$, (middle column) simulation model plus reference model discrepancy surrogate $\mathcal{M}+\boldsymbol{\delta}^\dagger$ and (right column) simulation model plus inferred model discrepancy surrogate $\mathcal{M} + \boldsymbol{\delta}$. Posterior mean (dots) $\pm 4\sigma$ (gray bar) of V_i and T_i are shown in contrast to the identity (red solid line) of the reference values $V_i^\dagger$ and $T_i\dagger$.

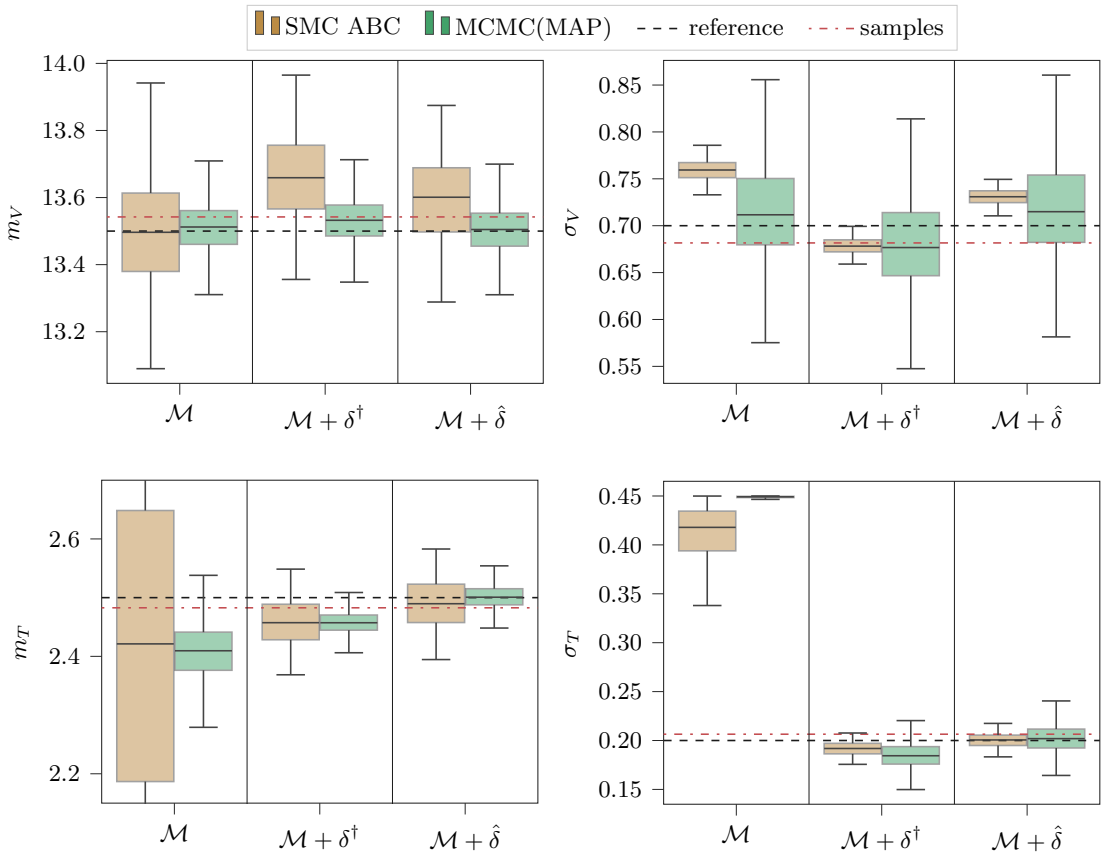


Figure 6.6.: Boxplots of the marginal posterior distributions of the hyper-parameters m_V, σ_V, m_T, σ_T obtained with SMC ABC and MCMC(MAP) for the test bench data. For each hyper-parameter results with (left column) only simulation model surrogate $\mathcal{M}$, (middle column) simulation model plus reference model discrepancy surrogate $\mathcal{M} + \boldsymbol{\delta}^\dagger$ and (right column) simulation model plus inferred model discrepancy surrogate $\mathcal{M}+\boldsymbol{\delta}$ are displayed together with the reference values and sample values.

discrepancy surrogate $\mathcal{M} + \boldsymbol{\delta}^{\dagger}$. This is surprising, as for construction of the inferred model discrepancy surrogate $\hat{\boldsymbol{\delta}}$ some non-optimal, w.r.t. reference model parameters, decisions for truncation parameter K_i were made.

Unfortunately, inference results with SMC ABC, albeit improving on m_T and σ_T, seem to deteriorate slightly for m_V and σ_V. This might be because of the noise standard deviation estimation required for the ABC method with summary statistics. In the Discussion 5.2.4 we argued that for the test bench data the estimates based on the stationary time interval are an underestimation and contribute to overestimated parameter standard deviations σ_V, σ_T. This might serve as an explanation here as well. For future investigations, an improved noise estimation might be beneficial. For instance, based on the already available marginal posterior distributions of σ_I, σ_ω from the BM2 model discrepancy inference on $\mathcal{Y}_{\mathcal{I}}$.

Also future work might research the effects of subset selection with different objectives and subset sizes onto the inference results. Another open point is that we used the posterior mean of the parameters and of the model discrepancy in $\mathcal{D}$ and consequently neglect the posterior uncertainty in the construction of $\hat{\boldsymbol{\delta}}$. Properly reflecting this uncertainty, would yield higher uncertainty in the parameter posterior distributions obtained via hierarchical inference. To achieve this, a surrogate directly based on the posterior distributions of $(\boldsymbol{x}_i, \boldsymbol{\delta}_i(\boldsymbol{x}_i))$ is required, which must then also be usable for inference. This is a challenging task and requires further research. One idea in this direction is to use GPs as surrogate that map the parameter posterior mean to Gaussian approximations of the model discrepancy posterior distributions. The GP covariance then adds to the covariance of the noise for hierarchical inference. To reflect this covariance in the ABC summary statistics a 2-Wasserstein metric without diagonal covariance assumption, for instance, is required.

Chapter 7.

Conclusions

> A mind that is stretched by a new experience can never go back to its old dimensions.
>
> Oliver Wendell Holmes, Jr. (1841-1935)

This thesis develops and extends Bayesian methods for inference of deterministic and aleatoric simulation model parameters from noisy measurement data with explicit consideration of simulation model discrepancy and quantification of the associated uncertainties. Application to a complex industrial example – an electric motor with synthetic and real-world data from a test bench – shows the effectiveness and robustness of the methods. An important ingredient for all methods is a PCE surrogate, which allows sampling from the posterior distribution in the first place by drastically speeding up the model evaluations.

For inference of deterministic parameters and model discrepancy simultaneously, the proposed Bayesian method – BM2 in Section 4.3 – efficiently improves physical parameter estimation, while learning a good approximation of the a-priori unknown model discrepancy. For the proposed iterative model discrepancy approximation with orthogonal functions, a smoothness assumption introducing specific prior knowledge is important to handle the underlying identifiability problem. Generally, the results show that a discrepancy model complexity well above the optimal one, leads to uncertain posterior distributions, indicating identifiability problems. This issue is circumvented by the proposed iterative model complexity determination. It properly selects an optimal complexity for the synthetic examples and equipped with an additional stopping criterion it also works for real-world data. Additionally to the already addressed points in the Discussion 4.5.4, evidence approximation might be an alternative criterion to select an optimal model complexity, and will be considered in future work for comparison purpose.

For inference of aleatoric parameters, both introduced methods – the hierarchical surrogate-based MCMC method and the novel hierarchical surrogate-based ABC method with summary statistics – show good inference results of comparable accuracy. Beneficial with ABC is the speed-up of the inference by exploiting parametric assumptions on the unknown parameter distributions and using summary statistics together with sparse grid quadrature. This is in particular effective for large data and high dimensional

parameter spaces. Albeit the proposed methodology has still some open points to work on, as discussed in Section 5.2.4, this worked showed that in cases where the likelihood is expensive to evaluate or even not available, ABC methods are an efficient way to obtain approximate posterior distributions with comparable quality in much less time. Additionally, Theorem 3 and experiments confirm posterior consistency for the surrogate-based ABC method.

The bias in the hierarchical inference results for the real-world test bench data (observed with both methods) motivates the combination of aleatoric parameter estimation with model discrepancy estimation. For this combined estimation the novel hierarchical surrogate-based discrepancy-corrected inference method is successful applied to the electric motor test bench data. The suggested divergence-based subset selection and PCE surrogate modeling for the model discrepancy provide an computationally efficient estimation of the unknowns. As a combination and extension of the previous methods it inherits their potential for improvements. Furthermore, as already discussed in Section 6.2.3, future work might address additional consideration of the subsets solution posterior uncertainty in the construction of the model discrepancy surrogate. As this is one of the first methods in this field and albeit its general concept, application to other complex problems will point out further directions for improvement.

Overall, this work showed that with an additional term for model discrepancy inference results can be improved and realistic information about the discrepancy can be collected, provided that at least some information about the parameters or the discrepancy is specified in the priors.

Generally, the combination of simulation models with data-driven approaches, such as the presented model discrepancy methods in this thesis, already have and will further gain increased importance in future scientific and industrial research. They can be seen as tools to discover weaknesses of simulation models, gain new insights and help to develop an improved next generation of a model. One might use the term ”data-driven model discovery”. With the methods developed in this thesis for black-box simulation models, insights are already available for model inputs and outputs. Opening the model and adding model discrepancy terms directly into the model equations might yield deeper insights and might simplify physical interpretation, at the potential cost of loosing general applicability of the method. Still remaining with this intrusive methods, are problems of identifiability, model selection and computational costs. Thus, the insights gained in this thesis can also be used for further research in that direction.

Appendix A.

Additional results for Chapter 4

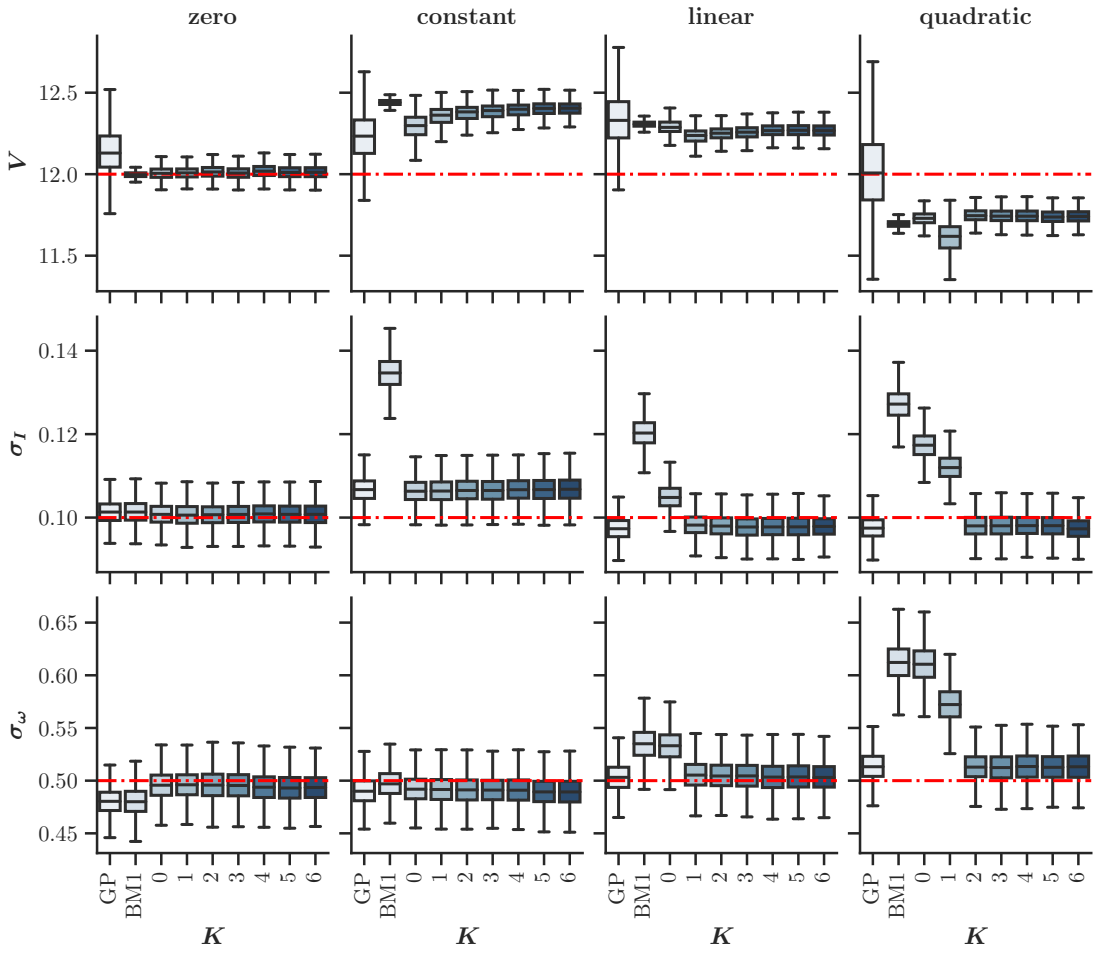


Figure A.1.: Posterior distributions of V, σ_I and σ_ω (rows) for discrepancy cases zero, constant, linear and quadratic (columns). Results are obtained via GP, BM1 and BM2 for $K = 0, \ldots, 6$ (with $b = 0.1$). Reference values are red dash-dotted.

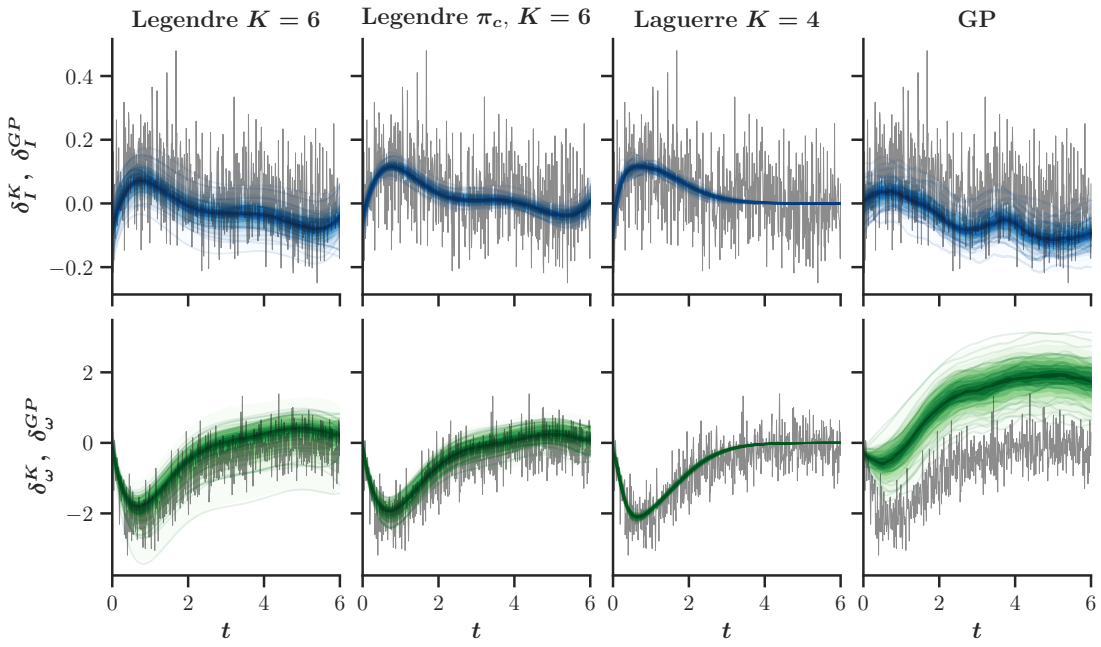


Figure A.2.: Discrepancy case J posterior distributions of δ_I, δ_ω (rows) for inference of V, T, σ_I and σ_ω (see Figure 4.11). Results are obtained via BM2 with Legendre polynomials (1st column with default prior and 2nd column with more informative, centered prior $\pi_c(V, T)$ for the parameters), weighted Laguerre polynomials (3rd column) and K specified in the heading (all with $b = 1$) and GP (4th column). Reference discrepancy plus measurement noise is gray.

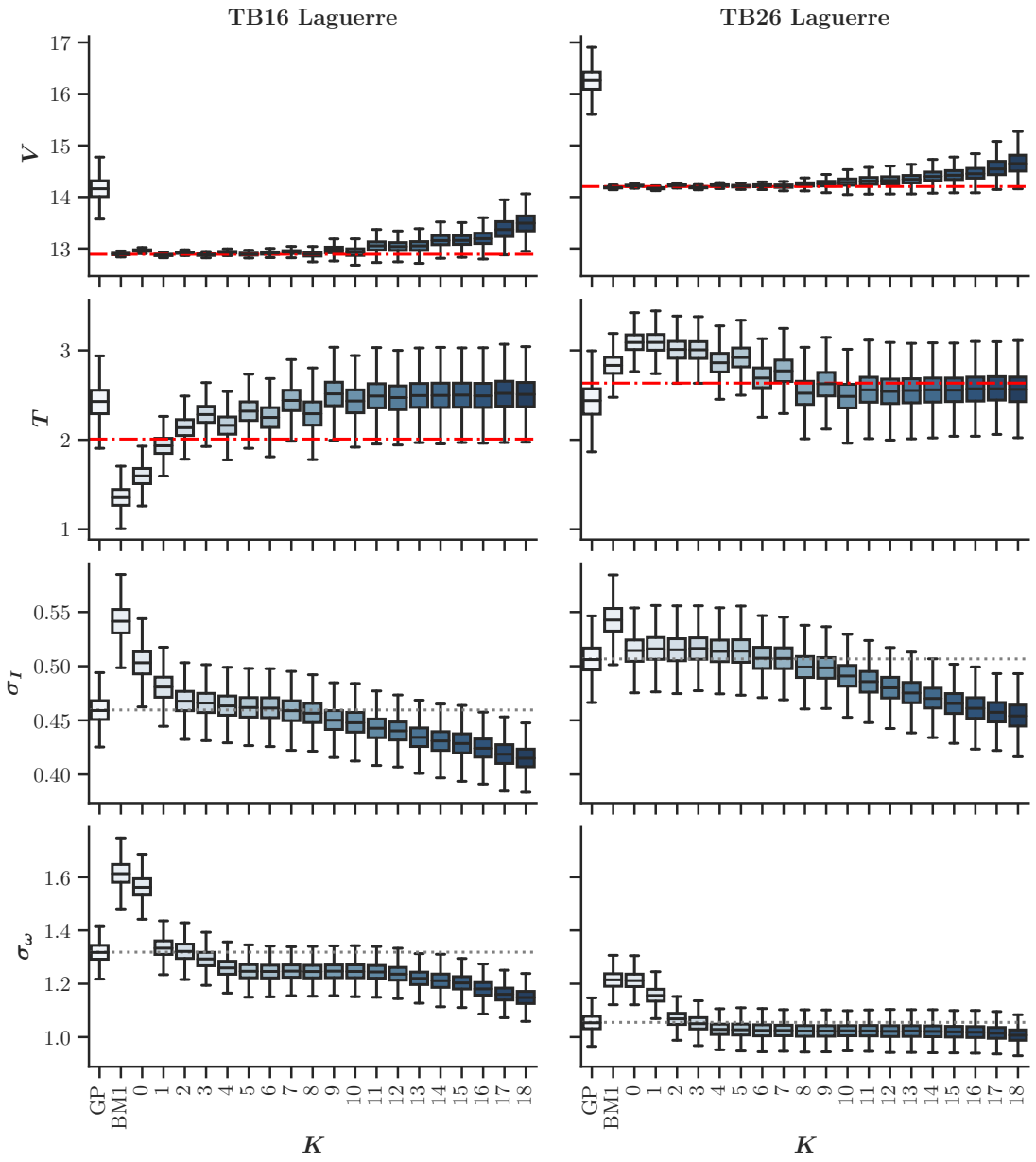


Figure A.3.: Test bench data TB16 and TB26 posterior distributions for V, T, σ_I and σ_ω (rows). Results are obtained via GP, BM1 and BM2 for $K = 0, \ldots, 18$ with weighted Laguerre polynomials. The coefficient prior is default $\pi(\boldsymbol{a}) = \mathit{Laplace}(0, b = 1)$. Reference values are red dash-dotted. Dotted gray lines augment the GP noise standard deviation posterior mean for comparison.

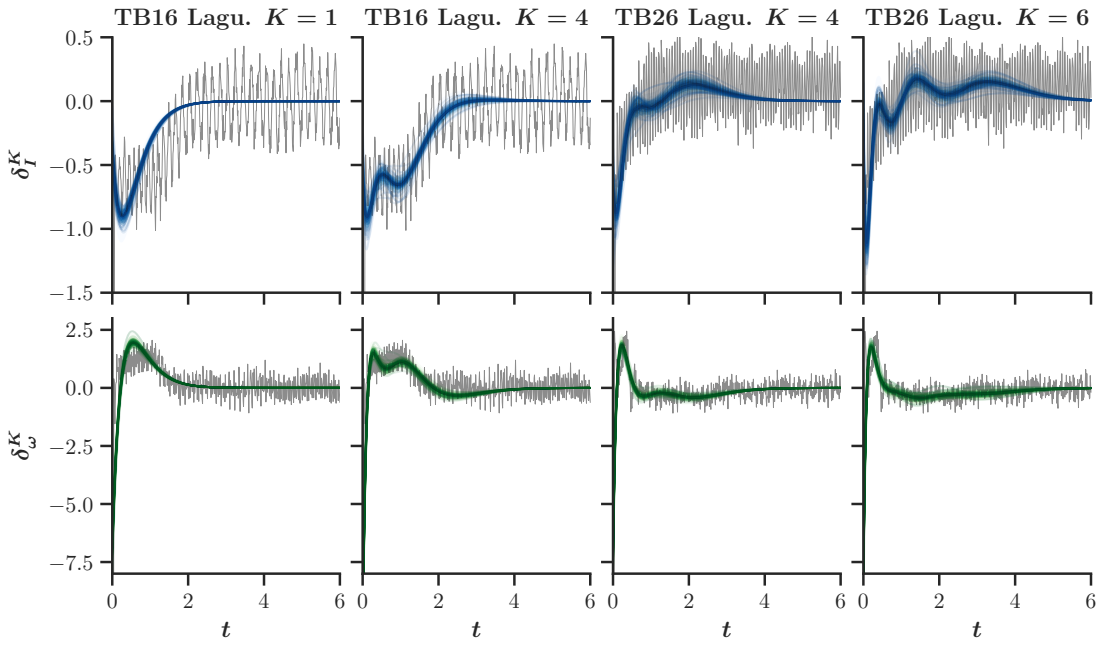


Figure A.4.: Posterior distributions of δ_I, δ_ω (rows) for inference of V, T, σ_I and σ_ω (see Figure A.3) for test bench data TB16 and TB26. Results are obtained via BM2 with weighted Laguerre polynomials with K specified in the heading (all with $b = 1$). Reference discrepancy plus measurement noise is gray.

Bibliography

[Abbati et al., 2019] Abbati, G., Wenk, P., Bauer, S., Osborne, M. A., Krause, A., and Schölkopf, B. (2019). AReS and MaRS-Adversarial and MMD-Minimizing Regression for SDEs. *arXiv preprint arXiv:1902.08480*.

[Ardizzone et al., 2019] Ardizzone, L., Kruse, J., Rother, C., and Köthe, U. (2019). Analyzing Inverse Problems with Invertible Neural Networks. In *International Conference on Learning Representations*.

[Arendt et al., 2012a] Arendt, P. D., Apley, D. W., and Chen, W. (2012a). Quantification of Model Uncertainty: Calibration, Model Discrepancy, and Identifiability. *Journal of Mechanical Design*, 134(10):100908.

[Arendt et al., 2012b] Arendt, P. D., Apley, D. W., Chen, W., Lamb, D., and Gorsich, D. (2012b). Improving Identifiability in Model Calibration Using Multiple Responses. *Journal of Mechanical Design*, 134(10):100909.

[Barber and Wang, 2014] Barber, D. and Wang, Y. (2014). Gaussian processes for Bayesian estimation in ordinary differential equations. In *International Conference on Machine Learning*, pages 1485–1493.

[Barber et al., 2015] Barber, S., Voss, J., and Webster, M. (2015). The rate of convergence for approximate Bayesian computation. *Electron. J. Statist.*, 9(1):80–105.

[Barnes et al., 2012] Barnes, C. P., Filippi, S., Stumpf, M. P., and Thorne, T. (2012). Considerate approaches to constructing summary statistics for ABC model selection. *Statistics and Computing*, 22(6):1181–1197.

[Bayes, 1763] Bayes, T. (1763). An essay toward solving a problem in the doctrine of chance. *Philosophical Transactions of the Royal Society of London*, 53:370–418.

[Beaumont et al., 2009] Beaumont, M. A., Cornuet, J.-M., Marin, J.-M., and Robert, C. P. (2009). Adaptive approximate Bayesian computation. *Biometrika*, 96(4):983—-990.

[Beaumont et al., 2002] Beaumont, M. A., Zhang, W., and Balding, D. J. (2002). Approximate Bayesian Computation in Population Genetics. *Genetics*, 162(4):2025–2035.

[Beck and Au, 2002] Beck, J. L. and Au, S.-K. (2002). Bayesian updating of structural models and reliability using Markov chain Monte Carlo simulation. *Journal of Engineering Mechanics*, 128(4):380–391.

[Beck and Muto, 2007] Beck, J. L. and Muto, M. (2007). Bayesian updating and model class selection of deteriorating hysteretic structural models using seismic response data. In *Proceedings of ECCOMAS Thematic Conference on Computational Methods in Structural Dynamics and Earthquake Engineering,*.

[Berveiller et al., 2006] Berveiller, M., Sudret, B., and Lemaire, M. (2006). Stochastic finite element: a non intrusive approach by regression. *European Journal of Computational Mechanics/Revue Européenne de Mécanique Numérique*, 15(1-3):81–92.

[Bhat et al., 2017] Bhat, K. S., Mebane, D. S., Mahapatra, P., and Storlie, C. B. (2017). Upscaling uncertainty with dynamic discrepancy for a multi-scale carbon capture system. *Journal of the American Statistical Association*, 112(520):1453–1467.

[Bishop, 2006] Bishop, C. M. (2006). *Pattern Recognition and Machine Learning*. Springer-Verlag, Berlin, Heidelberg.

[Blatman and Sudret, 2011] Blatman, G. and Sudret, B. (2011). Adaptive sparse polynomial chaos expansion based on least angle regression. *Journal of Computational Physics*, 230(6):2345 – 2367.

[Blum et al., 2013] Blum, M. G. B., Nunes, M. A., Prangle, D., and Sisson, S. A. (2013). A Comparative Review of Dimension Reduction Methods in Approximate Bayesian Computation. *Statist. Sci.*, 28(2):189–208.

[Bogachev, 1998] Bogachev, V. I. (1998). *Gaussian measures*. American Mathematical Soc.

[Bonilla et al., 2007] Bonilla, E. V., Chai, K. M., and Williams, C. (2007). Multi-task Gaussian Process Prediction. In Platt, J. C., Koller, D., Singer, Y., and Roweis, S. T., editors, *Advances in Neural Information Processing Systems 20*, pages 153–160. Curran Associates, Inc.

[Box and Draper, 1987] Box, G. E. and Draper, N. R. (1987). *Empirical model-building and response surfaces*, volume 424. Wiley New York.

[Box et al., 2009] Box, G. E., Luceno, A., and del Carmen Paniagua-Quinones, M. (2009). *Statistical control by monitoring and adjustment*. John Wiley & Sons.

[Bruder and Koutsourelakis, 2018] Bruder, L. and Koutsourelakis, P.-S. (2018). Beyond black-boxes in Bayesian inverse problems and model validation: applications in solid mechanics of elastography. *International Journal for Uncertainty Quantification*, 8(5).

[Brynjarsdóttir and O'Hagan, 2014] Brynjarsdóttir, J. and O'Hagan, A. (2014). Learning about physical parameters: The importance of model discrepancy. *Inverse Problems*, 30(11):114007.

[Bungartz and Griebel, 2004] Bungartz, H.-J. and Griebel, M. (2004). Sparse grids. *Acta numerica*, 13:147–269.

[Butterworth, 1930] Butterworth, S. (1930). On the theory of filter amplifiers. *Wireless Engineer*, 7(6):536–541.

[Calderhead et al., 2008] Calderhead, B., Girolami, M., and Lawrence, N. (2008). Accelerating Bayesian inference over nonlinear differential equations with Gaussian processes. In *Advances in Neural Information Processing Systems (NIPS)*.

[Cam, 1964] Cam, L. L. (1964). Sufficiency and Approximate Sufficiency. *The Annals of Mathematical Statistics*, 35(4):1419–1455.

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

[Chen et al., 2018] Chen, T. Q., Rubanova, Y., Bettencourt, J., and Duvenaud, D. K. (2018). Neural Ordinary Differential Equations. In Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., and Garnett, R., editors, *Advances in Neural Information Processing Systems 31*, pages 6572–6583. Curran Associates, Inc.

[Ching and Chen, 2007] Ching, J. and Chen, Y.-C. (2007). Transitional markov chain monte carlo method for bayesian model updating, model class selection, and model averaging. *Journal of engineering mechanics*, 133(7):816–832.

[Claeskens and Hjort, 2008] Claeskens, G. and Hjort, N. L. (2008). *Model Selection and Model Averaging*. Cambridge Books. Cambridge University Press.

[Conrad et al., 2017] Conrad, P. R., Girolami, M., Särkkä, S., Stuart, A., and Zygalakis, K. (2017). Statistical analysis of differential equations: introducing probability measures on numerical solutions. *Statistics and Computing*, 27(4):1065–1082.

[Conrad and Marzouk, 2013] Conrad, P. R. and Marzouk, Y. M. (2013). Adaptive Smolyak pseudospectral approximations. *SIAM Journal on Scientific Computing*, 35(6):A2643–A2670.

[Constantine et al., 2012] Constantine, P. G., Eldred, M. S., and Phipps, E. T. (2012). Sparse pseudospectral approximation method. *Computer Methods in Applied Mechanics and Engineering*, 229-232:1–12.

[Conti et al., 2009] Conti, S., Gosling, J. P., Oakley, J. E., and O'Hagan, A. (2009). Gaussian process emulation of dynamic computer codes. *Biometrika*, 96(3):663–676.

[Dashti et al., 2013] Dashti, M., Law, K. J. H., Stuart, A. M., and Voss, J. (2013). MAP estimators and their consistency in Bayesian nonparametric inverse problems. *Inverse Problems*, 29(9):095017.

[Dashti and Stuart, 2017] Dashti, M. and Stuart, A. M. (2017). The Bayesian Approach to Inverse Problems. In Ghanem, R., Higdon, D., and Owhadi, H., editors, *Handbook of Uncertainty Quantification*, pages 311–428. Springer International Publishing, Cham.

[Del Moral et al., 2006] Del Moral, P., Doucet, A., and Jasra, A. (2006). Sequential Monte Carlo samplers. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 68(3):411–436.

[Dondelinger et al., 2013] Dondelinger, F., Husmeier, D., Rogers, S., and Filippone, M. (2013). ODE parameter inference using adaptive gradient matching with Gaussian processes. In *Artificial Intelligence and Statistics (AISTATS)*, pages 216–228.

[Doppelbauer, 2018] Doppelbauer, M. (2018). The invention of the electric motor. [Online; Karlsruhe Instiute of Technology, Institute of Electrical Engineering, https://www.eti.kit.edu/english/1376.php and https://www.eti.kit.edu/english/1390.php; accessed Jan. 14, 2020].

[Draper and Smith, 1998] Draper, N. R. and Smith, H. (1998). *Applied regression analysis*, volume 326. John Wiley & Sons.

[Dunlop, 2019] Dunlop, M. M. (2019). Multiplicative noise in Bayesian inverse problems: Well-posedness and consistency of MAP estimators. *arXiv preprint arXiv:1910.14632*.

[Dunlop et al., 2017] Dunlop, M. M., Iglesias, M. A., and Stuart, A. M. (2017). Hierarchical Bayesian level set inversion. *Statistics and Computing*, 27(6):1555–1584.

[Edeling et al., 2014a] Edeling, W., Cinnella, P., and Dwight, R. (2014a). Predictive RANS simulations via Bayesian Model-Scenario Averaging. *Journal of Computational Physics*, 275:65 – 91.

[Edeling et al., 2014b] Edeling, W., Cinnella, P., Dwight, R., and Bijl, H. (2014b). Bayesian estimates of parameter variability in the k–ε turbulence model. *Journal of Computational Physics*, 258:73 – 94.

[Fearnhead and Prangle, 2012] Fearnhead, P. and Prangle, D. (2012). Constructing summary statistics for approximate Bayesian computation: semi-automatic approximate Bayesian computation. *Journal of the Royal Statistical Society Series B*, 74(3):419–474.

[Gelman et al., 2013] Gelman, A., Carlin, J. B., Stern, H. S., Dunson, D. B., Vehtari, A., Rubin, D. B., and Stern, H. S. (2013). *Bayesian Data Analysis*. Texts in Statistical Science Series. CHAPMAN & HALL/CRC and CRC Press, Boca Raton, 3rd edition.

[Gelman et al., 2014] Gelman, A., Hwang, J., and Vehtari, A. (2014). Understanding predictive information criteria for Bayesian models. *Statistics and Computing*, 24(6):997–1016.

[Ghanem and Spanos, 2003] Ghanem, R. G. and Spanos, P. D. (2003). *Stochastic finite elements: a spectral approach*. Courier Corporation.

[Giles, 2015] Giles, M. B. (2015). Multilevel Monte Carlo methods. *Acta Numerica*, 24:259–328.

[Giraldi et al., 2015] Giraldi, L., Liu, D., Matthies, H. G., and Nouy, A. (2015). To be or not to be intrusive? The solution of parametric and stochastic equations—Proper Generalized Decomposition. *SIAM Journal on Scientific Computing*, 37(1):A347–A368.

[Girolami et al., 2019] Girolami, M., Gregory, A., Yin, G., and Cirak, F. (2019). The Statistical Finite Element Method. *arXiv preprint arXiv:1905.06391*.

[Glaser, 2020] Glaser, P. (to appear, 2020). *Uncertainty Quantification for Complex Engineering Systems*. Dissertation, Heidelberg University.

[Glaser et al., 2017] Glaser, P., Kosmas, P., Schick, M., and Heuveline, V. (2017). Modeling of a Likelihood Function based on a Global Sensitivity Analysis. *PAMM*, 17(1):719–720.

[Glaser et al., 2016] Glaser, P., Schick, M., Petridis, K., and Heuveline, V. (2016). Comparison beween a Polynomial Chaos Surrogate Model and Markov Chain Monte Carlo for Inverse Uncertainty Quantification based on an Electric Drive Test Bench. *ECOMAS Congress*, 19:8809–8826.

[Gorbach et al., 2017] Gorbach, N. S., Bauer, S., and Buhmann, J. M. (2017). Scalable variational inference for dynamical systems. In *Advances in Neural Information Processing Systems*, pages 4806–4815.

[Goulet and Smith, 2013] Goulet, J.-A. and Smith, I. F. (2013). Structural identification with systematic errors and unknown uncertainty dependencies. *Computers & Structures*, 128:251 – 258.

[Gratiet et al., 2016] Gratiet, L. L., Marelli, S., and Sudret, B. (2016). Metamodel-based sensitivity analysis: polynomial chaos expansions and Gaussian processes. *Handbook of Uncertainty Quantification*, pages 1–37.

[Gu and Wang, 2018] Gu, M. and Wang, L. (2018). Scaled Gaussian stochastic process for computer model calibration and prediction. *SIAM/ASA Journal on Uncertainty Quantification*, 6(4):1555–1583.

[Guyon and Elisseeff, 2003] Guyon, I. and Elisseeff, A. (2003). An introduction to variable and feature selection. *Journal of machine learning research*, 3(Mar):1157–1182.

[Han et al., 2018] Han, J., Jentzen, A., and Weinan, E. (2018). Solving high-dimensional partial differential equations using deep learning. *Proceedings of the National Academy of Sciences*, 115(34):8505–8510.

[Hastie et al., 2009] Hastie, T., Tibshirani, R., and Friedman, J. H. (2009). *The elements of statistical learning: Data mining, inference, and prediction*. Springer series in statistics. Springer New York and Springer, New York, NY, 2nd edition.

[Hastings, 1970] Hastings, W. K. (1970). Monte Carlo Sampling Methods Using Markov Chains and Their Applications. *Biometrika*, 57(1):97.

[Helin and Burger, 2015] Helin, T. and Burger, M. (2015). Maximum a posteriori probability estimates in infinite-dimensional Bayesian inverse problems. *Inverse Problems*, 31(8):085009.

[Hennig et al., 2015] Hennig, P., Osborne, M. A., and Girolami, M. (2015). Probabilistic numerics and uncertainty in computations. *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, 471(2179).

[Higdon et al., 2004] Higdon, D., Kennedy, M., Cavendish, J. C., Cafeo, J. A., and Ryne, R. D. (2004). Combining field data and computer simulations for calibration and prediction. *SIAM Journal on Scientific Computing*, 26(2):448–466.

[Hoffman and Gelman, 2014] Hoffman, M. D. and Gelman, A. (2014). The No-U-turn sampler: adaptively setting path lengths in Hamiltonian Monte Carlo. *Journal of Machine Learning Research*, 15(1):1593–1623.

[Hu et al., 2017] Hu, Z., Yao, Z., and Li, J. (2017). On an adaptive preconditioned Crank–Nicolson MCMC algorithm for infinite dimensional Bayesian inference. *Journal of Computational Physics*, 332:492 – 503.

[Islam et al., 2004] Islam, M. S., Mir, S., and Sebastian, T. (2004). Issues in reducing the cogging torque of mass-produced permanent-magnet brushless dc motor. *IEEE Transactions on Industry Applications*, 40(3):813–820.

[John, 2016] John, D. (2016). *Uncertainty quantification for hydraulic systems with focus on the Smolyak sparse pseudo-spectral projection method*. Master thesis, Karlsruhe Institute of Technology (KIT).

[John et al., 2019a] John, D., Heuveline, V., and Schober, M. (2019a). GOODE: A Gaussian Off-The-Shelf Ordinary Differential Equation Solver. In Chaudhuri, K. and Salakhutdinov, R., editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 3152–3162, Long Beach, California, USA. PMLR.

[John et al., 2018a] John, D., Schick, M., and Heuveline, V. (2018a). Bayesian inference for estimating model discrepancy of an electric motor. *PAMM*, 18(1):e201800393.

[John et al., 2018b] John, D., Schick, M., and Heuveline, V. (2018b). Learning model discrepancy of an electric motor with Bayesian inference. *Preprint Series of the Engineering Mathematics and Computing Lab*, 0(01).

[John et al., 2020] John, D., Stohrer, L., Schillings, C., Schick, M., and Heuveline, V. (2020). Hierarchical surrogate-based Approximate Bayesian Computation for an electric motor test bench. In Review at Journal of Computational Physics.

[John et al., 2019b] John, D. N., Schick, M., and Heuveline, V. (2019b). Learning model discrepancy of an electric motor with Bayesian inference. In *Multidisciplinary Digital Publishing Institute Proceedings*, volume 33, page 11.

[Jung and Marjoram, 2011] Jung, H. and Marjoram, P. (2011). Choice of summary statistic weights in approximate Bayesian computation. *Statistical applications in genetics and molecular biology*, 10(1).

[Kaipio and Somersalo, 2005] Kaipio, J. and Somersalo, E. (2005). *Statistical and Computational Inverse Problems*. Applied Mathematical Sciences. Springer Science+Business Media, Inc, New York, NY.

[Kanagawa et al., 2018] Kanagawa, M., Hennig, P., Sejdinovic, D., and Sriperumbudur, B. K. (2018). Gaussian processes and kernel methods: A review on connections and equivalences. *arXiv preprint arXiv:1807.02582*.

[Kennedy et al., 2006] Kennedy, M. C., Anderson, C. W., Conti, S., and O'Hagan, A. (2006). Case studies in Gaussian process modelling of computer codes. *Reliability Engineering & System Safety*, 91(10):1301 – 1309. The Fourth International Conference on Sensitivity Analysis of Model Output (SAMO 2004).

[Kennedy and O'Hagan, 2001] Kennedy, M. C. and O'Hagan, A. (2001). Bayesian calibration of computer models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 63(3):425–464.

[Kersting et al., 2020] Kersting, H., Krämer, N., Schiegg, M., Daniel, C., Tiemann, M., and Hennig, P. (2020). Differentiable Likelihoods for Fast Inversion of'Likelihood-Free'Dynamical Systems. *arXiv preprint arXiv:2002.09301*.

[Kirkpatrick et al., 1983] Kirkpatrick, S., Gelatt, C. D., and Vecchi, M. P. (1983). Optimization by simulated annealing. *science*, 220(4598):671–680.

[Kotz et al., 2012] Kotz, S., Kozubowski, T., and Podgorski, K. (2012). *The Laplace distribution and generalizations: a revisit with applications to communications, economics, engineering, and finance*. Springer Science & Business Media.

[Lam et al., 2017] Lam, H., Zhang, X., and Plumlee, M. (2017). Improving prediction from stochastic simulation via model discrepancy learning. In *2017 Winter Simulation Conference (WSC)*, pages 1808–1819.

[Laplace, 1812] Laplace, P. S. (1812). Théorie analytique des probabilités, 2 vols. *Paris: Courcier Imprimeur*.

[Latz et al., 2019] Latz, J., Eisenberger, M., and Ullmann, E. (2019). Fast sampling of parameterised Gaussian random fields. *Computer Methods in Applied Mechanics and Engineering*, 348:978–1012.

[Le Maître and Knio, 2010] Le Maître, O. and Knio, O. M. (2010). *Spectral methods for uncertainty quantification: with applications to computational fluid dynamics*. Springer Science & Business Media.

[Lenz, 1834] Lenz, E. (1834). Ueber die bestimmung der richtung der durch elektrodynamische vertheilung erregten galvanischen ströme. *Annalen der Physik*, 107(31):483–494.

[Li, 2015] Li, J. (2015). A note on the Karhunen–Loève expansions for infinite-dimensional Bayesian inverse problems. *Statistics & Probability Letters*, 106:1 – 4.

[Li and Marzouk, 2014] Li, J. and Marzouk, Y. M. (2014). Adaptive construction of surrogates for the Bayesian solution of inverse problems. *SIAM Journal on Scientific Computing*, 36(3):A1163–A1186.

[Lintusaari et al., 2016] Lintusaari, J., Gutmann, M. U., Dutta, R., Kaski, S., and Corander, J. (2016). Fundamentals and Recent Developments in Approximate Bayesian Computation. *Systematic Biology*, 66(1):e66–e82.

[Lintusaari et al., 2018] Lintusaari, J., Vuollekoski, H., Kangasrääsiö, A., Skytén, K., Järvenpää, M., Marttinen, P., Gutmann, M., Vehtari, A., Corander, J., and Kaski, S. (2018). ELFI: Engine for Likelihood Free Inference. *arXiv preprint arXiv:1708.00707*.

[Macdonald et al., 2015] Macdonald, B., Higham, C. F., and Husmeier, D. (2015). Controversy in mechanistic modelling with Gaussian processes. In *International Conference on Machine Learning (ICML)*, volume 54, page 70.

[Mallasto and Feragen, 2017] Mallasto, A. and Feragen, A. (2017). Learning from uncertain curves: The 2-Wasserstein metric for Gaussian processes. In *Advances in Neural Information Processing Systems*, pages 5660–5670.

[Marin et al., 2012] Marin, J.-M., Pudlo, P., Robert, C. P., and Ryder, R. J. (2012). Approximate Bayesian computational methods. *Statistics and Computing*, 22(6):1167–1180.

[Marjoram et al., 2003] Marjoram, P., Molitor, J., Plagnol, V., and Tavaré, S. (2003). Markov chain Monte Carlo without likelihoods. *Proceedings of the National Academy of Sciences*, 100(26):15324–15328.

[Marzouk and Najm, 2009] Marzouk, Y. M. and Najm, H. N. (2009). Dimensionality reduction and polynomial chaos acceleration of Bayesian inference in inverse problems. *Journal of Computational Physics*, 228(6):1862 – 1902.

[McClarren, 2018] McClarren, R. G. (2018). *Uncertainty Quantification and Predictive Computational Science*. Springer.

[McGrayne, 2011] McGrayne, S. B. (2011). *The theory that would not die: how Bayes' rule cracked the enigma code, hunted down Russian submarines, & emerged triumphant from two centuries of controversy*. Yale University Press.

[Miller, 2002] Miller, A. (2002). *Subset selection in regression*. CRC Press.

[Morrison et al., 2018] Morrison, R. E., Oliver, T. A., and Moser, R. D. (2018). Representing model inadequacy: A stochastic operator approach. *SIAM/ASA Journal on Uncertainty Quantification*, 6(2):457–496.

[Nagel et al., 2017] Nagel, J. B., Rieckermann, J., and Sudret, B. (2017). Uncertainty quantification in urban drainage simulation: Fast surrogates for sensitivity analysis and model calibration. *arXiv preprint arXiv:1709.03283*.

[Nagel et al., 2020] Nagel, J. B., Rieckermann, J., and Sudret, B. (2020). Principal component analysis and sparse polynomial chaos expansions for global sensitivity analysis and model calibration: Application to urban drainage simulation. *Reliability Engineering & System Safety*, 195:106737.

[Najm and Chowdhary, 2016] Najm, H. and Chowdhary, K. (2016). Inference Given Summary Statistics. In Ghanem, R., Higdon, D., and Owhadi, H., editors, *Handbook of Uncertainty Quantification*, pages 1–35. Springer International Publishing, Cham.

[Neal, 2001] Neal, R. M. (2001). Annealed importance sampling. *Statistics and computing*, 11(2):125–139.

[Niu et al., 2016] Niu, M., Rogers, S., Filippone, M., and Husmeier, D. (2016). Fast inference in nonlinear dynamical systems using gradient matching. *Proceedings of Machine Learning Research*, 48:1699–1707.

[Nobile et al., 2008] Nobile, F., Tempone, R., and Webster, C. G. (2008). A Sparse Grid Stochastic Collocation Method for Partial Differential Equations with Random Input Data. *SIAM Journal on Numerical Analysis*, 46(5):2309–2345.

[Nunes and Balding, 2010] Nunes, M. A. and Balding, D. J. (2010). On Optimal Selection of Summary Statistics for Approximate Bayesian Computation. *Statistical Applications in Genetics and Molecular Biology*, 9(1).

[Oberkampf and Roy, 2010] Oberkampf, W. L. and Roy, C. J. (2010). *Verification and validation in scientific computing*. Cambridge University Press.

[O'Hagan, 2013] O'Hagan, A. (2013). Polynomial chaos: A tutorial and critique from a statistician's perspective. *SIAM/ASA J. Uncertainty Quantification*, 20:1–20.

[O'Hagan, 2019] O'Hagan, A. (2019). Expert Knowledge Elicitation: Subjective but Scientific. *The American Statistician*, 73(sup1):69–81.

[Ohm, 1826] Ohm, G. (1826). *Bestimmung des Gesetzes, nach welchem Metalle die Contaktelektricität leiten: nebst einem Entwurfe zur einer Theorie des Voltaischen Apparates und des Schweiggerschen Multiplicators*. Journal für Chemie und Physik.

[Parish and Duraisamy, 2016] Parish, E. J. and Duraisamy, K. (2016). A paradigm for data-driven predictive modeling using field inversion and machine learning. *Journal of Computational Physics*, 305:758 – 774.

[Pasquier and Smith, 2015] Pasquier, R. and Smith, I. F. (2015). Robust system identification and model predictions in the presence of systematic uncertainty. *Advanced Engineering Informatics*, 29(4):1096 – 1109. Collective Intelligence Modeling, Analysis, and Synthesis for Innovative Engineering Decision Making Special Issue of the 1st International Conference on Civil and Building Engineering Informatics.

[Paulo et al., 2012] Paulo, R., García-Donato, G., and Palomo, J. (2012). Calibration of computer models with multivariate output. *Computational Statistics & Data Analysis*, 56(12):3959–3974.

[Peherstorfer et al., 2018] Peherstorfer, B., Willcox, K., and Gunzburger, M. (2018). Survey of multifidelity methods in uncertainty propagation, inference, and optimization. *Siam Review*, 60(3):550–591.

[Plumlee, 2017] Plumlee, M. (2017). Bayesian Calibration of Inexact Computer Models. *Journal of the American Statistical Association*, 112(519):1274–1285.

[Plumlee, 2019] Plumlee, M. (2019). Computer model calibration with confidence and consistency. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 81(3):519–545.

[Plumlee and Lam, 2016] Plumlee, M. and Lam, H. (2016). Learning stochastic model discrepancy. In *2016 Winter Simulation Conference (WSC)*, pages 413–424.

[Plumlee and Lam, 2017] Plumlee, M. and Lam, H. (2017). An Uncertainty Quantification Method for Inexact Simulation Models. *arXiv preprint arXiv:1707.06544*.

[Popper, 2005] Popper, K. R. (2005). *The logic of scientific discovery*. Routledge.

[Prangle et al., 2014] Prangle, D., Fearnhead, P., Cox, M. P., Biggs, P. J., and French, N. P. (2014). Semi-automatic selection of summary statistics for ABC model choice. *Statistical Applications in Genetics and Molecular Biology*, 13(1):67–82.

[Quarteroni et al., 2010] Quarteroni, A., Sacco, R., and Saleri, F. (2010). *Numerical mathematics*, volume 37. Springer Science & Business Media, New York.

[Quiñonero-Candela and Rasmussen, 2005] Quiñonero-Candela, J. and Rasmussen, C. E. (2005). A unifying view of sparse approximate Gaussian process regression. *Journal of Machine Learning Research*, 6(Dec):1939–1959.

[Rackauckas et al., 2020] Rackauckas, C., Ma, Y., Martensen, J., Warner, C., Zubov, K., Supekar, R., Skinner, D., and Ramadhan, A. (2020). Universal Differential Equations for Scientific Machine Learning. *arXiv preprint arXiv:2001.04385*.

[Raissi et al., 2019] Raissi, M., Perdikaris, P., and Karniadakis, G. (2019). Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational Physics*, 378:686 – 707.

[Rasmussen and Williams, 2006] Rasmussen, C. E. and Williams, C. K. (2006). *Gaussian process for machine learning*. MIT press.

[Reich and Cotter, 2015] Reich, S. and Cotter, C. (2015). *Probabilistic forecasting and Bayesian data assimilation*. Cambridge University Press.

[Robert, 2007] Robert, C. (2007). *The Bayesian choice: from decision-theoretic foundations to computational implementation*. Springer Science & Business Media.

[Robert and Casella, 2005] Robert, C. P. and Casella, G. (2005). *Monte Carlo Statistical Methods*. Springer-Verlag, Berlin, Heidelberg.

[Roininen et al., 2019] Roininen, L., Girolami, M., Lasanen, S., and Markkanen, M. (2019). Hyperpriors for matérn fields with applications in bayesian inversion. *Inverse Problems & Imaging*, 13(1):1–29.

[Roy and Oberkampf, 2011] Roy, C. J. and Oberkampf, W. L. (2011). A comprehensive framework for verification, validation, and uncertainty quantification in scientific computing. *Computer Methods in Applied Mechanics and Engineering*, 200(25):2131 – 2144.

[Rudolf and Sprungk, 2018] Rudolf, D. and Sprungk, B. (2018). On a Generalization of the Preconditioned Crank–Nicolson Metropolis Algorithm. *Foundations of Computational Mathematics*, 18(2):309–343.

[Saltelli et al., 2008] Saltelli, A., Ratto, M., Andres, T., Campolongo, F., Cariboni, J., Gatelli, D., Saisana, M., and Tarantola, S. (2008). *Global sensitivity analysis: the primer*. John Wiley & Sons.

[Salvatier et al., 2016] Salvatier, J., Wiecki, T. V., and Fonnesbeck, C. (2016). Probabilistic programming in Python using PyMC3. *PeerJ Computer Science*, 2(2):e55.

[Sargsyan et al., 2019] Sargsyan, K., Huan, X., and Najm, H. (2019). Embedded model error representation for Bayesian model calibration. *International Journal for Uncertainty Quantification*, 9(4):365–394.

[Schillings and Schwab, 2016] Schillings, C. and Schwab, C. (2016). Scaling limits in computational Bayesian inversion. *ESAIM: Mathematical Modelling and Numerical Analysis*, 50(6):1825–1856.

[Schillings et al., 2019] Schillings, C., Sprungk, B., and Wacker, P. (2019). On the Convergence of the Laplace Approximation and Noise-Level-Robustness of Laplace-based Monte Carlo Methods for Bayesian Inverse Problems. *arXiv preprint arXiv:1901.03958*.

[Schober et al., 2019] Schober, M., Särkkä, S., and Hennig, P. (2019). A probabilistic model for the numerical solution of initial value problems. *Statistics and Computing*, 29(1):99–122.

[Scott, 2015] Scott, D. W. (2015). *Multivariate density estimation: theory, practice, and visualization*. John Wiley & Sons.

[Söderström and Stoica, 1989] Söderström, T. and Stoica, P. (1989). *System Identification*. Prentice-Hall, Inc. Online version (2001): http://www.it.uu.se/research/syscon/Ident.

[Simoen et al., 2013] Simoen, E., Papadimitriou, C., and Lombaert, G. (2013). On prediction error correlation in Bayesian model updating. *Journal of Sound and Vibration*, 332(18):4136 – 4152.

[Sinsbeck and Nowak, 2017] Sinsbeck, M. and Nowak, W. (2017). Sequential Design of Computer Experiments for the Solution of Bayesian Inverse Problems. *SIAM/ASA Journal on Uncertainty Quantification*, 5(1):640–664.

[Sisson et al., 2018] Sisson, S. A., Fan, Y., and Beaumont, M. (2018). *Handbook of approximate Bayesian computation*. Chapman and Hall/CRC.

[Sisson et al., 2007] Sisson, S. A., Fan, Y., and Tanaka, M. M. (2007). Sequential Monte Carlo without likelihoods. *Proceedings of the National Academy of Sciences*, 104(6):1760–1765.

[Skilling, 2006] Skilling, J. (2006). Nested sampling for general Bayesian computation. *Bayesian analysis*, 1(4):833–859.

[Smith, 2013] Smith, R. C. (2013). *Uncertainly Quantification: Theory, Implementation, and Applications*, volume 12. SIAM Computational Science & Engineering Series, Philadelphia, PA, USA.

[Sobol, 2001] Sobol, I. M. (2001). Global sensitivity indices for nonlinear mathematical models and their Monte Carlo estimates. *Mathematics and computers in simulation*, 55(1-3):271–280.

[Sobol, 2003] Sobol, I. M. (2003). Theorems and examples on high dimensional model representation. *Reliability Engineering & System Safety*, 79(2):187–193.

[Solak et al., 2003] Solak, E., Murray-smith, R., Leithead, W. E., Leith, D. J., and Rasmussen, C. E. (2003). Derivative Observations in Gaussian Process Models of Dynamic Systems. In Becker, S., Thrun, S., and Obermayer, K., editors, *Advances in Neural Information Processing Systems 15*, pages 1057–1064. MIT Press.

[Sprungk, 2018] Sprungk, B. (2018). *Numerical Methods for Bayesian Inference in Hilbert Spaces*. Universitätsverlag der TU Chemnitz, Chemnitz.

[Sraj et al., 2016] Sraj, I., Le Maître, O. P., Knio, O. M., and Hoteit, I. (2016). Coordinate transformation and Polynomial Chaos for the Bayesian inference of a Gaussian process with parametrized prior covariance function. *Computer Methods in Applied Mechanics and Engineering*, 298:205–228.

[Stigler, 1986] Stigler, S. M. (1986). Laplace's 1774 memoir on inverse probability. *Statistical Science*, 1(3):359–363.

[Stine, 1923] Stine, W. M. (1923). *The Contributions of HFE Lenz to Electromagnetism*. Acorn Press.

[Stuart, 2010] Stuart, A. M. (2010). Inverse problems: A Bayesian perspective. *Acta Numerica*, 19:451–559.

[Sudret, 2008] Sudret, B. (2008). Global sensitivity analysis using polynomial chaos expansions. *Reliability engineering & system safety*, 93(7):964–979.

[Sullivan, 2015] Sullivan, T. J. (2015). *Introduction to Uncertainty Quantification*, volume 63 of *Texts in applied mathematics*. Springer.

[Sunnaker et al., 2013] Sunnaker, M., Busetto, A. G., Numminen, E., Corander, J., Foll, M., and Dessimoz, C. (2013). Approximate Bayesian Computation. *PLOS Computational Biology*, 9(1):1–10.

[Tavaré et al., 1997] Tavaré, S., Balding, D. J., Griffiths, R. C., and Donnelly, P. (1997). Inferring Coalescence Times From DNA Sequence Data. *Genetics*, 145(2):505–518.

[Theano Development Team, 2016] Theano Development Team (2016). Theano: A Python framework for fast computation of mathematical expressions. *arXiv preprint arXiv:1605.02688*.

[Titsias, 2009] Titsias, M. (2009). Variational learning of inducing variables in sparse Gaussian processes. In *Artificial Intelligence and Statistics*, pages 567–574.

[Toliyat and Kliman, 2004] Toliyat, H. A. and Kliman, G. B. (2004). *Handbook of electric motors*, volume 120 of *Electrical and computer engineering*. CRC press, New York and Basel, 2nd edition.

[Tuo, 2019] Tuo, R. (2019). Adjustments to computer models via projected kernel calibration. *SIAM/ASA Journal on Uncertainty Quantification*, 7(2):553–578.

[Tuo and Wu, 2015] Tuo, R. and Wu, C. F. J. (2015). Efficient calibration for imperfect computer models. *Ann. Statist.*, 43(6):2331–2352.

[Tuo and Wu, 2016] Tuo, R. and Wu, C. F. J. (2016). A Theoretical Framework for Calibration in Computer Models: Parametrization, Estimation and Convergence Properties. *SIAM/ASA Journal on Uncertainty Quantification*, 4(1):767–795.

[Tuzlukov, 2002] Tuzlukov, V. (2002). *Signal processing noise*. CRC Press.

[Uribe et al., 2020] Uribe, F., Papaioannou, I., Betz, W., and Straub, D. (2020). Bayesian inference of random fields represented with the Karhunen–Loève expansion. *Computer Methods in Applied Mechanics and Engineering*, 358:112632.

[Vehtari et al., 2017] Vehtari, A., Gelman, A., and Gabry, J. (2017). Practical Bayesian model evaluation using leave-one-out cross-validation and WAIC. *Statistics and computing*, 27(5):1413–1432.

[Wanner and Hairer, 1991] Wanner, G. and Hairer, E. (1991). *Solving ordinary differential equations I*, volume 1. Springer-Verlag, Berlin.

[Watanabe, 2010] Watanabe, S. (2010). Asymptotic equivalence of bayes cross validation and widely applicable information criterion in singular learning theory. *Journal of Machine Learning Research*, 11(Dec):3571–3594.

[Wenk et al., 2019a] Wenk, P., Abbati, G., Bauer, S., Osborne, M. A., Krause, A., and Schölkopf, B. (2019a). ODIN: ODE-Informed Regression for Parameter and State Inference in Time-Continuous Dynamical Systems. *arXiv preprint arXiv:1902.06278*.

[Wenk et al., 2019b] Wenk, P., Gotovos, A., Bauer, S., Gorbach, N., Krause, A., and Buhmann, J. M. (2019b). Fast Gaussian Process Based Gradient Matching for Parameter Identification in Systems of Nonlinear ODEs. In *Proceedings of the 22nd International Conference on Artificial Intelligence and Statistics (AISTATS)*, volume 89, pages 1351–1360. PMLR.

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

[Wilkinson, 2013] Wilkinson, R. D. (2013). Approximate Bayesian computation (ABC) gives exact results under the assumption of model error. *Statistical Applications in Genetics and Molecular Biology*, 12(2):129–141.

[Wilkinson et al., 2010] Wilkinson, R. D., Steiper, M. E., Soligo, C., Martin, R. D., Yang, Z., and Tavaré, S. (2010). Dating Primate Divergences through an Integrated Analysis of Palaeontological and Molecular Data. *Systematic Biology*, 60(1):16–31.

[Wong et al., 2017] Wong, R. K., Storlie, C. B., and Lee, T. C. (2017). A frequentist approach to computer model calibration. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 79(2):635–648.

[Xie and Xu, 2020] Xie, F. and Xu, Y. (2020). Bayesian Projected Calibration of Computer Models. *Journal of the American Statistical Association*, 0(0):1–18.

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

[Yu and Liu, 2004] Yu, L. and Liu, H. (2004). Efficient feature selection via analysis of relevance and redundancy. *Journal of machine learning research*, 5(Oct):1205–1224.

[Zhang et al., 2013] Zhang, Y., Li, S., Wang, T., and Zhang, Z. (2013). Divergence-based feature selection for separate classes. *Neurocomputing*, 101:32 – 42.