On Novel Approaches to Model-Based Structural Health Monitoring

A Thesis submitted to the University of Sheffield

for the degree of Doctor of Philosophy in the Faculty of Engineering

by

P. A. Gardner

Department of Mechanical Engineering

University of Sheffield

September 2018

Abstract

Structural health monitoring (SHM) strategies have classically fallen into two main categories of approach: model-driven and data-driven methods. The former utilises physics-based models and inverse techniques as a method for inferring the health state of a structure from changes to updated parameters; hence defined as inverse model-driven approaches. The other frames SHM within a statistical pattern recognition paradigm. These methods require no physical modelling, instead inferring relationships between data and health states directly. Although successes with both approaches have been made, they both suffer from significant drawbacks, namely parameter estimation and interpretation difficulties within the inverse model-driven framework, and a lack of available full-system damage state data for data-driven techniques. Consequently, this thesis seeks to outline and develop a framework for an alternative category of approach; forward model-driven SHM. This class of strategies utilise calibrated physics-based models, in a forward manner, to generate health state data (i.e. the undamaged condition and damage states of interest) for training machine learning or pattern recognition technologies. As a result the framework seeks to provide potential solutions to these issues by removing the need for making health decisions from updated parameters and providing a mechanism for obtaining health state data.

In light of this objective, a framework for forward model-driven SHM is established, highlighting key challenges and technologies that are required for realising this category of approach. The framework is constructed from two main components: generating physics-based models that accurately predict outputs under various damage scenarios, and machine learning methods used to infer decision bounds. This thesis deals with the former, developing technologies and strategies for producing

statistically representative predictions from physics-based models. Specifically this work seeks to define validation within this context and propose a validation strategy, develop technologies that infer uncertainties from various sources, including model discrepancy, and offer a solution to the issue of validating full-system predictions when data is not available at this level.

The first section defines validation within a forward model-driven context, offering a strategy of hypothesis testing, statistical distance metrics, visualisation tools, such as the witness function, and deterministic metrics. The statistical distances field is shown to provide a wealth of potential validation metrics that consider whole probability distributions. Additionally, existing validation metrics can be categorised within this fields terminology, providing greater insight.

In the second part of this study emulator technologies, specifically Gaussian Process (GP) methods, are discussed. Practical implementation considerations are examined, including the establishment of validation and diagnostic techniques. Various GP extensions are outlined, with particular focus on technologies for dealing with large data sets and their applicability as emulators. Utilising these technologies two techniques for calibrating models, whilst accounting for and inferring model discrepancies, are demonstrated: Bayesian Calibration and Bias Correction (BCBC) and Bayesian History Matching (BHM). Both methods were applied to representative building structures in order to demonstrate their effectiveness within a forward model-driven SHM strategy. Sequential design heuristics were developed for BHM along with an importance sampling based technique for inferring the functional model discrepancy uncertainties.

The third body of work proposes a multi-level uncertainty integration strategy by developing a subfunction discrepancy approach. This technique seeks to construct a methodology for producing valid full-system predictions through a combination of validated sub-system models where uncertainties and model discrepancy have been quantified. This procedure is demonstrated on a numerical shear structure where it is shown to be effective.

Finally, conclusions about the aforementioned technologies are provided. In addition, a review of the future directions for forward model-driven SHM are outlined with the hope that this category receives wider investigation within the SHM community.

Acknowledgements

First and foremost I would like express my sincere gratitude to my supervisor, Dr. Robert Barthorpe. I would like to thank him for making this work possible and for providing both his support and guidance throughout the last three years. Thanks is also attributed to my second supervisor, Dr. Charles Lord, for both his encouragement and advice. I am grateful for the opportunities you have both provided and for creating a positive atmosphere in which to conduct this research.

Secondly I am grateful for all those who have made the Dynamics Research Group in Sheffield such a friendly and interesting place to work. Specifically, I would like to thank Tim and Ramon for their friendship, encouragement and for insightful discussions throughout this work.

Also, I would like to thank my friends and family for their support. In particular, I am grateful for the opportunities and help my parents have provided over the years. Last, but no means least, I would like to thank my wife Charlotte. She has listened to more conversations on aspects of this work than anybody else, has been patient throughout the late nights of work and has been so supportive. Thanks, I owe you one.

Table of Contents

List of Acronyms

ABC	Approximate Bayesian Computation
ANN	Artificial Neural Networks
AR	Autoregressive
ARD	Automatic Relevance Determination
BCBC	Bayesian Calibration and Bias Correction
BHM	Bayesian History Matching
BIC	Bayesian Information Criterion
BLA	Bayes Linear Analysis
CDF	Cumulative Density Function
CFD	Computational Fluid Dynamics
DNN	Deep Neural Networks
DoE	Design of Experiments
DTC	Deterministic Training Conditional
ECDF	Empirical Cumulative Density Function
ES	Entropy Search
FEA	Finite Element Analysis
FITC	Fully Independent Training Conditional
GA	Genetic Algorithm
GMLHC	Generalised Maximum Latin Hypercube

GMLHD	Generalised Maximum Latin Hypercube Design
GP	Gaussian Process
GSA	Global Sensitivity Analysis
i.i.d.	independent and identically distributed
IPE	Individual Prediction Error
IPM	Integral Probability Metric
KDE	Kernel Density Estimate
KL	Kullback-Leibler
KNN	K Nearest Neighbours
KS	Kolmogorov-Smirnoff
KSD	Kernel Stein Discrepancy
LHC	Latin Hypercube
LHD	Latin Hypercube Design
MCMC	Markov Chain Monte Carlo
MLE	Maximum Likelihood Estimate
MMD	Maximum Mean Discrepancy
MSE	Mean Squared Error
NARX	Nonlinear Auto-Regressive eXogenous inputs
NLML	Negative Log Marginal Likelihood
NMSE	Normalised Mean Squared Error
p-value	probability value
PCE	Polynomial Chaos Expansion
PDF	Probability Density Function
PEP	Power Expectation Propagation
PES	Predictive Entropy Search
QQ-Plot	Quantile-Quantile Plot
RKHS	Reproducing Kernel Hilbert Space

RVM	Relevance Vector Machines
SE	Squared Exponential
SHM	Structural Health Monitoring
SMC	Sequential Monte Carlo
SoD	Subset of Data
SVM	Support Vector Machine
UP	Uncertainty Propagation
UQ	Uncertainty Quantification
V&V	Verification and Validation
VFE	Variational Free Energy

Chapter 1

Introduction

Although successes have been made in the field of Structural Health Monitoring (SHM) several key challenges still remain. These mainly revolve around the lack of available damage state data at a full-system level as well as problems in inferring and interpreting updated model parameters. For these reasons this thesis seeks to develop novel approaches to model-based SHM offering a strategy for performing forward model-driven SHM. In order to realise this goal key technologies and methodologies are developed.

1.1 Structural Health Monitoring: Objectives, Benefits and Challenges

SHM defines the implementation of an online process whereby data from a structure is acquired and interpreted in order to assess the health state of the structure. SHM technologies seek to provide early indications of damage occurrences in order to aid and inform asset management decisions with the broad aim of eliminating in-service failures and unscheduled maintenance. In the context of SHM damage is broadly defined as a change that adversely affects the structure's performance [1]. These SHM tools aim to allow operators to move towards a predictive maintenance strategy providing a variety of potential benefits. An SHM process provides economic benefits by reducing system failure, increasing maintenance efficiency, and providing monitoring data for cost-effective data-driven design. Implementing an SHM strategy

also improves safety whilst offering a methodology for extending the design life of a structure. These objectives and benefits of SHM are widely applicable across a variety of industries, for example, manufacturing, power generation, aerospace and civil infrastructure.

The particular tasks an SHM technology is required to performed can be divided into a hierarchy of levels as defined by Rytter [2]:

1. *Detection*: information about the presence of damage
2. *Location*: information about the position of the damage
3. *Classification*: information about the type of damage
4. *Assessment*: information about the extent of the damage
5. *Prognosis*: information about the residual life and safety of the structure

Progression to the next level in the hierarchy requires successful completion of the previous levels, where a greater level of desired detail results in an increased difficulty. It is well-established that prognosis is distinguished from the other levels, as it can only be achieved with an understanding of the damage physics [3]. Accordingly, a key decision in the implementation of an SHM strategy is that of identifying the level of identification required.

In the author's opinion there are generally three main challenges in developing robust SHM technologies. Firstly, damage cannot be directly measured [4]. This means that other quantities are collected with the expectation that they contain information about the health state of the structure. This information is extracted through a feature selection process in order to identify damage sensitive features, however this is currently a very bespoke procedure. Secondly, confounding influences, such as environmental conditions, changes in boundary conditions and/or loading obscure patterns in the data that are associated with damage. These must therefore be removed before health decision strategies are implemented. Thirdly, currently most SHM techniques require damage state data from all damage scenarios of interest, often in a range of operational conditions. Usually it is not feasible to obtain these data either because it is not economically viable, practical or would pose safety concerns. This thesis aims to provide a potential framework for resolving the lack of available data problem, but the strategy could also provide methods for solving the other two challenges.

1.2 Approaches to Structural Health Monitoring

Throughout SHM literature methods are divided into two categories of approach: model-driven and data-driven [4–7]. Model-driven (also known as physics-based) methods use law-based models in combination with inverse techniques in order to infer or 'update' a set of parameters [8, 9] — commonly referred to as model updating. Health decisions are then made through interpreting these updated parameter values, leading to the category of methods being herein defined as *inverse model-driven*. In contrast, data-driven methods seek to 'learn' relationships between measured response data and structural damage states based on pattern recognition or machine learning-based models; all without the construction of a physics-based model [1, 4, 5, 10]. Decisions about the health state of a structure are subsequently made via classifications (or more generally predictions) of in-service data through the inferred statistical model.

Further to these two well-established divisions, a third category of approach exists. This class is distinct from the previous two, as it combines physics-based models, utilised in a forward manner, and statistical pattern recognition methodologies [11]; herein defined as *forward model-driven* approaches. This thesis seeks to outline an overarching framework for forward model-driven approaches, comment on how this class provides potential solutions to issues with the existing two categories, develop technologies within the framework and finally to summarise areas of further research for realising forward model-driven SHM.

1.2.1 Inverse Model-Driven Approaches

Inverse model-driven techniques often involve the construction of a high-fidelity model of the structure, for which health decisions are to be made, typically in the form of a Finite Element Analysis (FEA) model. The procedure for making health decisions often follows a two step process. Initially, the model is calibrated so that it more accurately represents the structure in question. This is generally performed by model updating, based on in-service data of the undamaged condition. The second stage involves obtaining in-service monitoring data, for which the health state is unknown. The model is then updated again based on this in-service data and changes in the inferred model parameters from the baseline calibration are used to

perform damage identification at levels 1-4 of Rytters hierarchy. Prognosis may also be achievable because an updated physics-based model is generated through the inverse model-driven procedure [3].

SHM via an inverse model-driven approach therefore relies on model updating processes. Model updating refers to techniques where certain model parameters are adjusted such that the residual between observational data and model predictions is minimised [12]. This task is broadly attempted in two general approaches: direct methods, where structural matrices are updated to reproduce measured data, and sensitivity methods, where error between predictions and observations are minimised via changing a set of defined parameters [12, 13]. Commonly in SHM sensitivity based techniques are selected over direct approaches. This is because attempting to update full structural matrices within a direct approach often leads to a lack of control over the updated matrix values, leading to inferred parameters with little physical meaning.

Initial development of model updating methodologies approached the problem from a deterministic view, e.g. the well-established iterative sensitivity based method [14]. Such techniques approached the problem of model updating using optimisation technologies, whereby a cost function is developed, typically in a least squares formulation, and parameter steps made via sensitivity matrices [14, 15]. However, these approaches require regularisation due to the problem of model updating being ill-posed [9]. These deterministic methods also have difficulties in handling variability and uncertainties that are present, e.g. from environmental conditions, parametric variability and model form uncertainties. For these reasons alternative frameworks for approaching model updating have been developed.

Two popular philosophical approaches for handling uncertainties within model updating are fuzzy and Bayesian methods [16]. Fuzzy techniques are non-probabilistic approaches that transform uncertainties into fuzzy inputs, i.e. as a fuzzy number — a quantity that is characterised by a membership function — and then perform multiple optimisation problems [17]. Fuzzy model updating technologies assume that the fuzzy input variables are independent and equally likely, which will result in the worst case range of parameters being inferred. Bayesian methods, *per contra*, take a probabilistic view of parameter estimation, using Bayes' theorem (see Appendix A.1 for mathematical definitions and details) to update model parameters and their uncertainties. In certain scenarios these methods contain inherent model regularisation contained within the marginal likelihood, sometimes referred to as the Bayesian

Occam's razor [18]. Beck and Katafygiotis provide a review of Bayesian model updating [19, 20]. Nonetheless most of the current model updating methodologies fail to account for uncertainties associated with model form errors, known as model discrepancy. Failure to consider this form of uncertainty will often lead to bias in the estimated parameters, and therefore incorrect health statements.

Inverse model-driven technologies suffer from several challenges when implemented as part of an SHM strategy. Firstly, the type and number of parameters to use must be selected [21, 22]. In scenarios where damage is unknown (e.g. both in location and type), as is often the case, this can lead to an especially large number of parameters. Parametrisation becomes increasingly challenging as model fidelity increases, where there are a large number of potential parameter sets. Another difficulty is that of interpreting the updated parameters to make a decision about the structure's health. This can be especially difficult when parameters affect structural stiffness, as multiple phenomena influence changes in stiffness. An accurate understanding of the physics must inform whether updated parameters are no longer physically meaningful rather than altered by the presence of damage, and constraints placed on the updating process when this is the case. As mentioned, variability and uncertainties within the 'target' data must be handled as part of the updating process. Moreover, these issues are confounded by the problem that a solution, or a unique stable solution, for the inverse approach cannot always be achieved due to ill-conditioning. These non-identifiability issues become of increasing concern when the parameter values are being used for health diagnostics, as repeats of the update may lead to different conclusions.

1.2.2 Data-Driven Approaches

Data-driven methods approach SHM as a pattern recognition problem, where a statistical model is 'learnt' from a set of training observations from the structure and used to label new in-service data [4]. As the data sets are from the structure in operation, the complete loading environment is incorporated into establishing the normal, undamaged condition (and any other labelled classes). This category of approach removes the need for developing any physics-based models of the structure, relying solely on the information contained within the data, inherently capturing variations and uncertainties that the data contains.

A general framework for data-driven approaches involves the steps outlined as follows [4, 5]:

- **Sensing and data acquisition** — optimally located sensors acquire data from the structure.
- **Pre-processing** — data normalisation, cleaning, compression and fusion occur, with the aim of removing confounding influences, problems arising from the acquisition phase, reducing dimensionality and combining multiple information sources.
- **Feature extraction** — data is converted into damage sensitive features, quantities that clearly state the damage function to be learnt.
- **Post-processing** — features may need additional normalisation, cleaning, compression or fusion.
- **Machine Learning** — a classification, regression or density estimation algorithm is trained using the extracted damage sensitive features.
- **Decision** — new in-service data are provided to the machine learning method, analysed and a decision about the health state is made.

Crucial decisions for a data-driven approach are what features and machine learning method to utilise. The objective of any machine learning technology within SHM is to infer trends or functions such that the relationships define a normal condition, where a departure from the normal condition diagnoses damage. These techniques can be categorised into solutions for three main problems, stated in order of complexity [23, 24]:

- **Classification** - data are assigned labels based on an inferred decision bound.
- **Regression** - an unknown function is inferred based on an input-output mapping.
- **Density Estimation** - clusters of probability densities are inferred from data.

Further to these divisions, machine learning methods can be categorised as supervised, unsupervised and semi-supervised. Supervised and unsupervised algorithms are

distinguished by whether labels for data (e.g. the damage state of the structure) are known or unknown respectively. Semi-supervised learning combines both labelled and unlabelled data, usually where the latter is more numerous. Typically, regression and classification are supervised problems whereas density estimation is unsupervised. Furthermore, due to the absence of labelled data, unsupervised methods can only be used to perform novelty detection — the process of inferring whether a change has occurred. Once a difference has been detected, labels need to be obtained in order to make statements about what the change refers to, therefore requiring a level of supervision at this stage. On the other hand supervised methods can be used to perform levels 1-4 of Rytter's hierarchy. A variety of classification methods — such as Support Vector Machine (SVM) [24], Relevance Vector Machines (RVM) [25], and Artificial Neural Networks (ANN) [26] — regression methods — for example, Gaussian Process (GP)s [27] and ANNs [26] — and density estimation techniques — e.g. Gaussian mixture models, K Nearest Neighbours (KNN) and kernel density estimation — have been implemented within the SHM literature. For a review on machine learning methodologies implemented within SHM and their successes the reader is referred to [4, 28].

Challenges remain in implementing data-driven methods. Supervised approaches require in-service, labelled data from all damage states of interest in order to infer robust decision thresholds. This is often not economically viable or feasible at a full-system level, resulting in a significant challenge to their implementation. In addition, unsupervised techniques suffer from all the complexities of performing density estimation, as well as challenges in obtaining labels when in-service data appears outside the normal condition. Semi-supervised learning, although providing a degree of solution to these problems, still requires some level of labelled full-system data.

1.2.3 Forward Model-Driven Approaches

The third, and less established category of approach to SHM, is forward model-driven SHM. Here models are utilised in a forward manner, whereby their predictions form training data for supervised machine learning methods [11]. This class of technologies incorporates elements of both inverse model-driven and data-driven, where model calibration theory and machine learning techniques are combined. The motivation for developing forward model-driven approaches is to aid the challenges in obtaining

labelled damage state data, as well as removing complexities in inferring damage from parameter updates.

Few examples of forward model-driven approaches exist within the literature. FEA models have been used to generate features for ANNs in performing damage identification in bridges [29, 30]. Satpal et al. implemented a combined model updating and SVM approach where model predictions trained the classier [31], with Hariri-Ardebili and Pourkamali-Anaraki applying a similar methodology to concrete dams [32]. Most of these approaches utilise deterministic FEA model outputs, with a few adding arbitrary noise terms to replicate variability, whilst others propagate 'known' parameter uncertainties through Monte Carlo realisations. None of these methods consider model form errors, and either do not attempt to validate their models or implement full-system damage state data in the validation process. As a consequence these approaches fail to tackle the key challenges facing SHM technologies. This lack of thorough investigation within the literature and failure to address key SHM challenges provides the motivation for clearly outlining a forward model-driven framework, presenting the main difficulties and providing technological solutions to these issues.

1.3 Objectives

This thesis seeks to establish and develop a framework for forward model-driven SHM, providing a methodology for overcoming the aforementioned issues with current inverse model-driven and data-driven approaches to SHM. Contributions are made in developing specific technologies required for this category of approach, focusing on three main challenges:

- Defining and generating a validation procedure for forward model-driven SHM, requiring new validation metrics that consider complete probability distributions.
- Calibrating computer models under various sources of uncertainty, including model discrepancy — important for producing accurate forward predictions.
- Tackling issues associated with creating a validated full-system model when observational data is not obtainable at this level.

By outlining challenges to the proposed framework, providing potential research avenues and appropriate technologies, it is hoped that this category of approach will receive a wider uptake within the SHM community.

1.4 Chapter Summary

The outline of this thesis is as follows:

Chapter 2 — The proposed framework for performing forward model-driven SHM is presented. Motivation for the approach is discussed along with an introduction to key components and challenges to implementation, culminating in an explanation of the research objectives targeted by this thesis.

Chapter 3 — Validation within a forward model-driven context is discussed, whereby a validation strategy is determined. Specific validation metrics that consider complete probability distribution are defined and compared on numerical examples.

Chapter 4 — Due to the computational burden of evaluating a computer model, such as an FEA model, and the numerous runs required for most statistical and optimisation methods, computationally efficient emulators are outlined. Specifically Gaussian Process emulators are investigated, discussing issues of implementation and validation before describing extensions for large data sets, multiple outputs, stochastic computer models and dynamic processes.

Chapter 5 — The first of two chapters proposing methods for dealing with model discrepancy inference. A mathematical formulation for performing Bayesian Calibration and Bias Correction is defined before being applied to two representative building structure examples.

Chapter 6 — An alternative approximate Bayesian methodology for achieving calibration whilst accounting for model discrepancy is described, namely Bayesian History Matching. Extensions to the methodology are proposed with techniques for incorporating sequential design strategies, as well as functional inference of the model discrepancy via importance sampling. The technology is subsequently applied to a representative building structure case study.

Chapter 7 — The problem of validating a full-system model without health state data at this level is investigated. A multi-level uncertainty integration strategy using

a subfunction discrepancy approach is developed in which multiple sub-system level models are validated and their uncertainties propagated through to the full-system model. This technique is explored in a numerical case study of a shear structure under the introduction of reduced bolt tension and open cracking.

Chapter 8 — Themes and technologies presented throughout the thesis are brought together and conclusions are outlined. Merits of the proposed technologies and strategies are discussed along with challenges for implementation and their combination in a forward model-driven SHM framework. Finally areas of further research are detailed.

Chapter 2

Forward Model-Driven Structural Health Monitoring

SHM technologies that utilise physics-based models have often approached structural health diagnosis by using changes to inferred parameters as a method for making health statements, part of the inverse model-driven category of approaches. These techniques often suffer from non-identifiability issues, difficulties in parametrisation of the model and interpretation of the updated parameters. In contrast, forward model-driven SHM provides a framework whereby validated models, employed in a forward manner, generate predictions of damage sensitive features that are statistically representative of health state data obtained from the operational structure. The emphasis in this class of methods is that models can be used as a proxy in order to generate damage state data that would otherwise not be economically viable or practically infeasible to obtain from the in-service structure. These health state predictions from models are subsequently incorporated in training pattern recognition or machine learning classification technologies, which can be implemented online in order to make health diagnostic decisions.

As a result, forward model-driven SHM provides a solution to the lack of available damage state data problem within data-driven approaches. Furthermore, models offer additional tools for performing feature selection as well as the design of monitoring systems, i.e. the locations, type and positions of a particular sensor network *a priori* to implementation. Finally, the physics-based models developed in a forward model-driven approach offer a methodology for performing prognosis, achieving the

complete set of levels in Rytter's hierarchy.

This chapter formally outlines a framework for forward model-driven SHM, capturing the key procedures and technologies required to generate robust health state predictions from models. The framework is summarised in a flowchart providing a clear implementation strategy and a division of the methodology into research objectives. Components of the framework that offer significant benefits or require additional research for implementation are subsequently highlighted, providing motivation for the proceeding chapters.

2.1 A Framework for Forward Model-Driven Structural Health Monitoring

Forward model-driven SHM offers an alternative methodology for approaching the problem of SHM. The framework, built around utilising models (herein defined as *simulators*) in a forward manner, seeks to tackle challenges associated with both inverse model-driven and data-driven approaches. This is achieved by employing models in a forward manner, reducing many of the difficulties associated with the parametrisation and interpretation of updated parameters in inverse model-driven approaches and providing a practical and cost-effective technique for designing sensor networks, performing feature selection, obtaining health state data and achieving prognosis, all of which are challenges to data-driven methods.

Forward model-driven methods are comprised of two main components; generating representative damage state features from simulators, and using those predictions to train machine learning or pattern recognition approaches. The second component, well studied within the data-driven category of SHM, has been demonstrated to be effective when labelled damage state data is available, as outlined in Section 1.2.2. Within a forward model-driven approach these techniques generally remain mathematically and algorithmically the same, with the only difference arising from the source of training data, i.e. simulator based predictions. Consequently, the major challenges in establishing a forward model-driven strategy are in developing methodologies and technologies that achieve the objective of the first component, namely the generation of representative damage state features from a simulator.

Generating representative predictions from simulators means tackling several key

challenges. Firstly, there must be a method for determining whether simulator predictions of health states are representative of those obtained operationally. This requires the definition of what a valid simulator prediction is within the forward model-driven context. In order to develop this definition an understanding of how these prediction are used within classification methods must be established. In a data-driven framework, features extracted from operational data are often employed in training decision bounds that capture the expected behaviour of the particular damage feature under each damage scenario in the training set. This means that health state data generated from simulator predictions must capture the inherent variability and progression of the health state in question. A simulator will therefore be valid if its predictions generate statistical distributions of health states that are statistically similar to those obtained observationally. Consequently, it is the author's opinion that a non-deterministic philosophy is required to realise this goal.

Secondly, generating statistically representative predictions will involve some level of calibration and a validation procedure. Unfortunately, both these processes require data from the real-world structure leaving the conundrum of how to calibrate and validate the simulator given that health state data is neither feasible to obtain nor cost-effective in the majority of applications. If this question is not tackled, forward model-driven approaches simply become an expensive and demanding way to perform sub-standard data-driven SHM, introducing further approximations and modelling challenges. One solution to this problem is the division of the structure in question, and hence the simulator, into a set of components, sub-assembly etc., for which obtaining health state data is feasible and economically viable. In this scenario a full-system, such as a aeroplane, is divided into various sub-systems, e.g. wing panels, riveted joints, landing gear assemblies, coupons etc., where each sub-system can be tested under damage types which are expected to be likely causes of failure in the full-system. Small scale test strategies can then be developed, or existing certification tests used to collect data sets that can be implemented in calibrating and validating the set of simulators. The usefulness of forward model-driven technologies rest on the ability to utilise and integrate these sub-system data sets into calibrating and validating sub-system level simulators, which when propagated through to the full-system, via an algebra of simulators and uncertainty management, produce valid, i.e. statistically representative predictions, which have required no full-system health state data. Obviously this is an incredibly ambitious goal, nonetheless methods such as multi-level uncertainty integration strategies offer techniques for undertaking such a challenge.

Thirdly, procedures for calibrating simulators should involve mechanisms for handling multiple sources of uncertainty, especially those from model form errors, known as model discrepancy. Statistically representative predictions will not often be achievable without capturing observational variability, along with parameter uncertainties and accounting for any functional model discrepancy — the differences between simulator outputs and observational data. Accordingly, it is the author's opinion that calibration will be best achieved in a probabilistic framework where statistical models are constructed that have mechanisms for quantifying uncertainties from various sources, including observational variability, inherent stochasticity due to parameter uncertainties and model discrepancy.

The proposed forward model-driven SHM framework outlined within this chapter aims to capture the processes required to overcome these challenges. Figures 2.1 and 2.2 present a flowchart of the framework, showing the progression from sub-system analysis to full-system predictions and health state identification.

The flowchart in Figs. 2.1 and 2.2 is described as follows. Prior beliefs about types of damage the full-system may be subject to are identified and used to divide the structure into appropriate sub-systems; where these damage mechanisms can be captured. These divisions of the structure are then analysed to ensure that experimental data, required for calibrating and validating these sub-systems, is both practically and economically feasible. If this is not the case then divisions of the full-system are updated based on cost-analysis. Once acceptable sub-systems are identified the process moves to the sub-modelling phase.

For each sub-system simulators are constructed and validated in order to capture the relevant damage mechanisms. This process begins with defining prior assumptions about the simulator's model form based on the appropriate damage physics. This leads to simulator development where computer software (such as a FEA, Computational Fluid Dynamics (CFD) or multi-physics packages) are utilised or analytical models constructed such that the simulator captures the target process and damage mechanisms. Verification is performed to ensure that the numerical methods and approximations within the simulator behave as expected. If the simulator fails this verification process the simulator development stage is repeated, with an update to the model form. Following the simulator passing the verification processes, damage feature selection is performed. The simulator is utilised to explore potential outputs and their mathematical transforms sensitivity to the damage scenarios being modelled. The most sensitive damage feature(s) is/are then employed in developing an

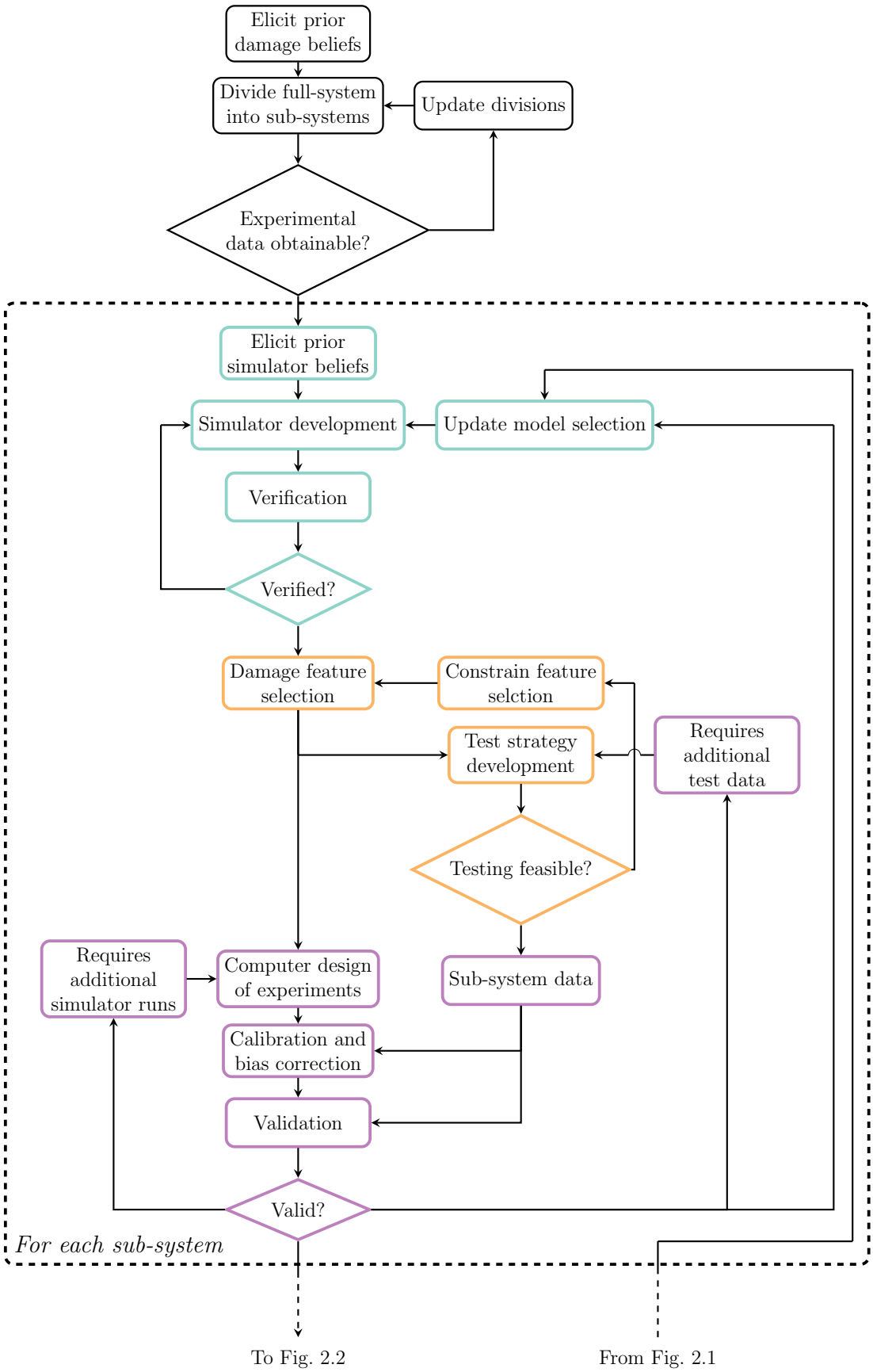


Figure 2.1: Flowchart of a forward model-driven framework — Part 1.

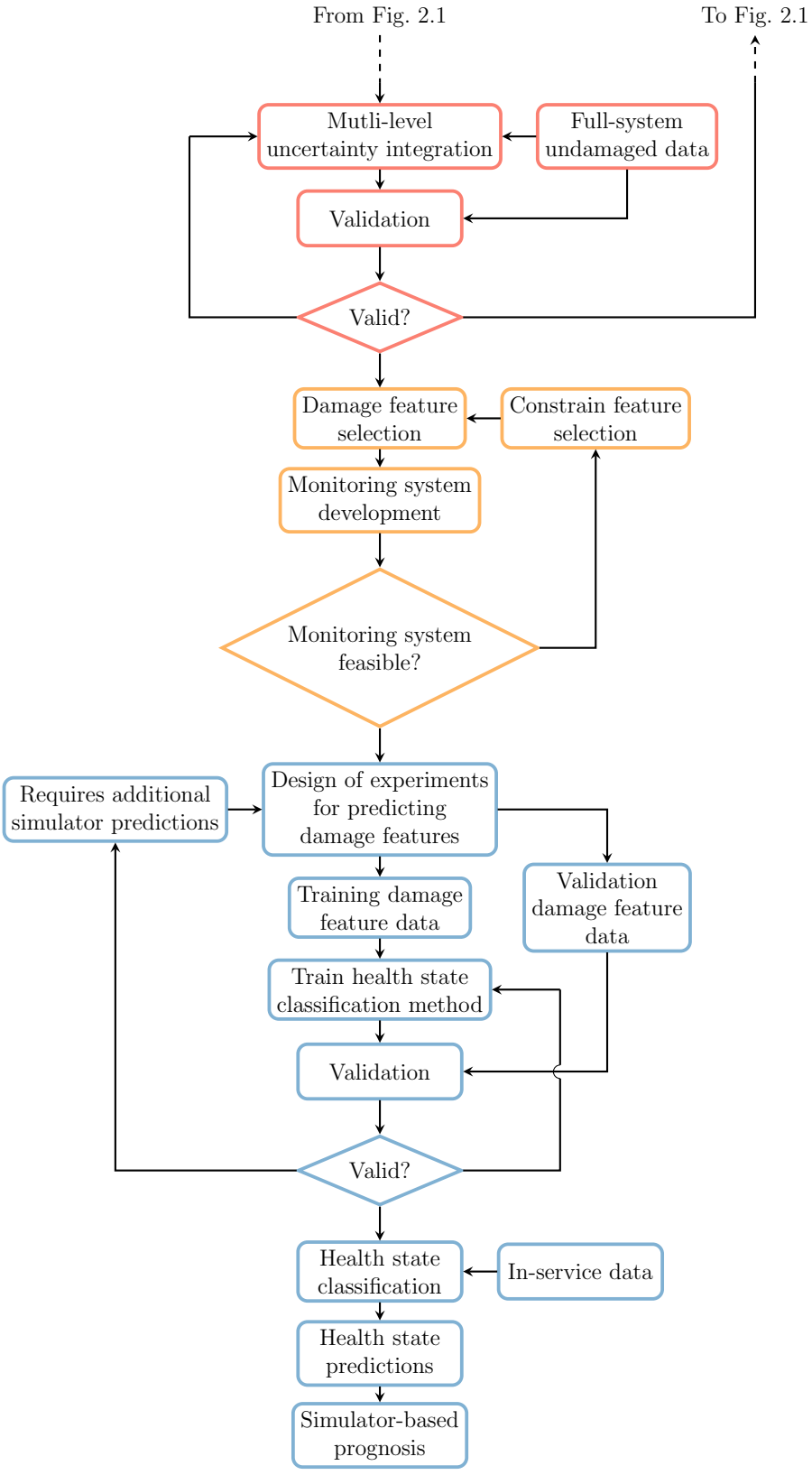


Figure 2.2: Flowchart of a forward model-driven framework — Part 2.

experimental test strategy. This process involves using the simulator to identify the locations, number and sensor types that can be employed in obtaining the calibration and validation data sets for the sub-system in question. The test strategy is assessed to ensure that it is cost-effective and practical, where failure may lead to the adoption of alternative damage features and/or test strategies. Proceeding acceptance of the test strategy, sub-system data is collected, generating both a calibration and validation set. At this time design of computer experiments are generated in order to provide simulator runs for the calibration and validation processes, capturing the expected input and parameter domains. The simulator evaluations and sub-system calibration data set are then used to infer the parameter distributions as well as the model discrepancy for the damage types being modelled. Subsequently, the calibrated and bias corrected simulator outputs are validated against an independent data set from the sub-system. Failure to pass the validation requirements will either lead to better defining the simulator input and parameter domain via additional simulator runs, the collection of more sub-system data, or in the worst case an update of the simulator through some improvement model selection. Which path to take requires a detailed decision process, left as an area of further research. If the particular sub-system simulator is determined to be valid then the next sub-system is constructed and validated until all the simulators are deemed valid. It is noted that some sub-system simulators will require inputs or parameters that are determined from other sub-system simulators within the chain.

After creating the required validated sub-system simulators a multi-level uncertainty integration process is employed. The simulator outputs and their quantified uncertainties are integrated to make predictions at a full-system level. At this point it may be possible to obtain full-system undamaged state data. This can be used to confirm and validate the model form of the complete full-system model — however the damage mechanisms are assumed to be captured and validated based on the sub-system modelling. If outputs from the multi-level uncertainty integration scheme are not valid then the complete full-system model is updated, and the algebra between simulators and their uncertainties amended.

The successive tasks, after generating a valid full-system model from the multi-level uncertainty integration strategy, are to identify and select damage sensitive features from the model. Next these selected damage features are used to develop a monitoring system, calculating locations, number and types of sensors required to ensure an appropriate probability of detecting the hypothesised damage scenarios.

The proposed monitoring system can be assessed in order to determine whether it is practical and cost-effective, otherwise alternative damage features and/or monitoring systems are proposed.

The next processes involve generating training and validation data of the selected damage features from the full-system model. This will involve a design of computer experiments in order to cover the phenomena of interest. A chosen classification technology is subsequently trained using the training data sets and validated using the independent validation set. The process is repeated, updating the classification methods parameters, or acquiring training data from the full-system model until the inferred classification method is deemed valid.

The SHM monitoring system can now be employed online, once the sensing infrastructure is deployed. As in-service data is collected it is transformed into the appropriate feature space before being classified based on the inferred classification bounds. These predictions will identify the damage location, types and extent informing which sub-system simulator combination to use in performing simulator-based prognosis. The acquisition of in-service data can also be used to improve the full-system model, where additional model selection and inferences can be performed to increase accuracy of predictions and reduce modelling uncertainties.

Based on the flowchart in Figs. 2.1 and 2.2 the framework has several main elements:

- **Model Development and Selection** (*green*) - using prior beliefs about a structure and the processes to be modelled, in order to select an appropriate simulator that captures the model form at the required level of fidelity. Model selection specifically defines the process of using observational data to select the most appropriate simulator.

- **Damage Feature Selection and Monitoring System Design** (*orange*) - the ability to use a simulator to investigate potential output quantities and mathematical transforms that are sensitive to the onset of particular damage scenarios. Monitoring system design is performed by utilising the simulator to explore the measurement type, number and location of sensors before experimental or in-service data are acquired.

- **Simulator Calibration (and Validation)** (*purple*) - the ability to infer system parameters, model discrepancy and all associated uncertainties intro-

duced in the modelling and data acquisition processes via inverse Uncertainty Quantification (UQ) methods; where validation is performed probabilistically.

- **Multi-Level Uncertainty Integration** (*red*) - the ability to use sub-system level observational data to calibrate and validate sub-system level simulators, which can be combined in a manner such that valid full-system predictions are made without the need for full-system health state data.

- **Health State Decision Strategies** (*blue*) - machine learning or pattern recognition methods used to infer decision bounds (as studied within data-driven SHM research).

The proceeding sections outline these five key elements in more detail, highlight challenges, and potential technologies and methodologies for performing each task.

2.1.1 Model Development and Selection

Model development is the process of building a simulator, either through computer software, such as FEA, or numerical modelling, that captures the behaviour of a given physical process. In the context of forward model-driven SHM this involves creating a simulator that models particular damage events. To achieve valid simulator predictions, research will need to go into developing more accurate mathematical damage models.

The definition of model selection refers to the process of selecting the most appropriate model from a set of candidate models given a data set. This often involves comparing model evidences, i.e. the probability of the data $\boldsymbol{z}$ given a particular model $\mathcal{M}_i$, $p(\boldsymbol{z} \mid \mathcal{M}_i)$, such as used in Bayesian Information Criterion (BIC) [33].

Model development and selection are challenging problems that ultimately rest on the ability to define appropriate candidate models. Forward model-driven SHM also provides the added complication in that although model development and selection may be done on a sub-system level, the model that explains the sub-system data best may not be the model that helps improve full-system predictions the most. This difficulty means that additional research is required in analysing the trade-offs between full-system predictive capability and appropriateness of the selected model based on sub-system data.

2.1.2 Damage Feature Selection and Monitoring System Design

A clear benefit of forward model-driven SHM is that by developing simulators of the structure various outputs and their mathematical transforms can be investigated as potential damage sensitive features. Furthermore, once a particular damage feature has been selected, the simulator(s) can be utilised in selecting and optimising a sensor network all before physical testing or in-service data has been collected. This offers significant cost benefits and risk reduction as monitoring setups can be considered virtually.

One approach to damage feature selection using simulators is via sensitivity analysis techniques, specifically Global Sensitivity Analysis (GSA) which aims to determine the variation of an output quantity in terms of the variation in the inputs [34–36]. This would lead to an assessment of the sensitivity of a given set of outputs and their transforms to changes in the inputs, specifically the extent and location of each particular damage type being considered. In addition, the proposed features can be assessed for sensitivity to other inputs, aiming to identify a feature that is sensitive to damage alone, rather than other confounding influences.

Monitoring system design, i.e. selecting the number, location and type of sensors to implement on a structure, has been attempted using a variety of methods, e.g. energetic techniques [37], information [37, 38] and risk based approaches [39]. A positive by-product of forward model-driven SHM is that the availability of simulators means that these techniques become applicable within the framework.

2.1.3 Calibration

The success of a forward model-driven framework relies on the ability to generate validated damage features that are statistically representative of those obtained in-service. As a consequence, calibration is vital for capturing the behaviour of the structure in operational conditions under different damage scenarios and in producing robust health decisions. Various calibration methodologies exist and have been implemented within inverse model-driven SHM, usually under the term of model updating, as mentioned in Section 1.2.1. However, many of these approaches fail to incorporate a mechanism for including model discrepancy, a phenomena that

exists due to the fact that simulators contain simplifications or the absence of certain physics leading to a mismatch between observational data and simulator prediction even when the 'true' parameters are known. This section aims to demonstrate why the exclusion of a belief that model discrepancy exists within a calibration process will lead to misidentified parameter distributions, as well as poor predictive performance, due to the simulator's model form errors.

Model Discrepancy

The importance of incorporating a mechanism to account for, and infer, model discrepancy within a calibration procedure is demonstrated on a numerical example — a mass, tensioned wire system. Figure 2.3 illustrates a mass, tension wired system with a centred and off-centred mass, for which the natural frequency can be calculated using Eqs. (2.1) and (2.2) respectively.

$$w_n = \frac{1}{\pi}\sqrt{\frac{T}{Ml}} \tag{2.1}$$

$$w_n = \frac{1}{2\pi}\sqrt{\frac{T(a+b)}{M(ab)}} \tag{2.2}$$

Where M is the mass in kg, T is the tension in N, l is the length in m between the fixed boundaries, a and b are the offset distances (where $a = l - b$) and ω_n is the natural frequency in Hz. For the numerical examples parameter values are set for $l = 1$, $a = 0.2$ and the observational mass $\hat{M} = 5.43$kg. The aim of calibration in this numerical study is to find the posterior mass distribution given observations of the natural frequency at ten equally spaced tensions from 200-1000N.

In the first scenario, no model discrepancy is present. This means that the simulator has captured all physics that govern the behaviour of natural frequency for a reduction in tension (for the mass, tensioned wire system). This leads to the simulator having a functional form defined in Eq. (2.2). Consequently the observations are also mathematically governed by Eq. (2.2) but with the addition of observational uncertainty $e \sim \mathcal{N}(0, 0.01)$.

In the second scenario model discrepancy is introduced. Here the simulator models

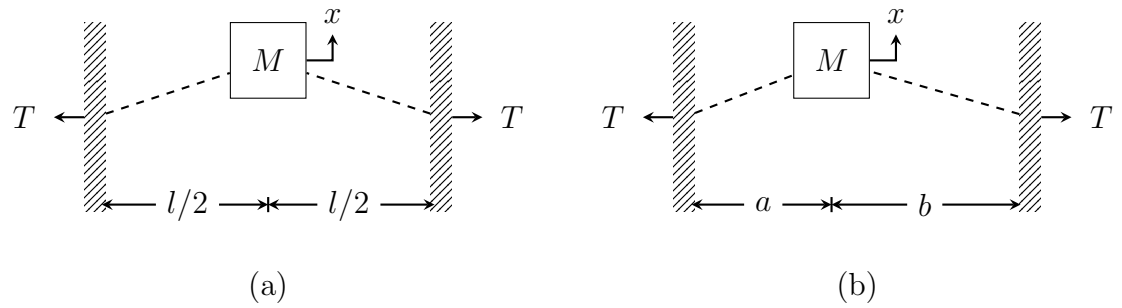


Figure 2.3: Mass, tensioned wire system. Panel (a) presents a centred mass and panel (b) an off-centred mass.

the mass, tensioned wire system as being fixed in the centre, representing a level of missing physics in the problem. The observations are the same as obtained in scenario one, i.e. from Eq. (2.2) with additive Gaussian noise.

For this numerical example Bayesian calibration is implemented (see Appendix A.1 for more details on Bayes' theorem) via Markov Chain Monte Carlo (MCMC), specifically using a Metropolis-Hastings random walk algorithm as outlined in Algorithm 1. The prior was $p(\theta) = \mathcal{N}(5, 1)$, where θ is the parameter being calibrated, which in this example is the mass M. A Gaussian likelihood function is implemented, $p(\boldsymbol{z} \,|\, \theta) = \mathcal{N}(\boldsymbol{z} \,|\, f(\boldsymbol{x}, \theta), \sigma_n^2)$. The likelihood contains the simulator function $f(\boldsymbol{x}, \theta)$, given inputs $\boldsymbol{x}$ which here are tensions, parameters θ which is a given mass and a fixed noise variance $\sigma_n^2 = 0.01$, the observations are denoted as $\boldsymbol{z}$. For both scenarios the proposal variance was $V = 0.01$ where $100,000$ posterior samples were generated after a $10,000$ sample burn in (i.e. the first $10,000$ accepted samples are thrown away to ensure the states of the Markov chain have entered a region of high probability). The resulting Markov chains were checked for convergence.

Figure 2.4 presents the outcomes of calibrating the two scenarios via MCMC sampling. Figure 2.4c displays the inferred posterior distributions of the mass for both scenarios one and two, with a comparison of the 'true' parameter value 5.43kg. This demonstrates that when the simulator and observational data come from the same mathematical functions (i.e. there are no simplifications or missing physics) calibration is achievable, reflected in the mean of the distribution closely matching that of the true parameter, and the parameter distribution being clearly centred around that value. However, when model form errors are present, as in scenario two, the parameter distribution shifts away from the 'true' parameter (i.e. bias is introduced); in this example almost no probability mass is located near the 'true'

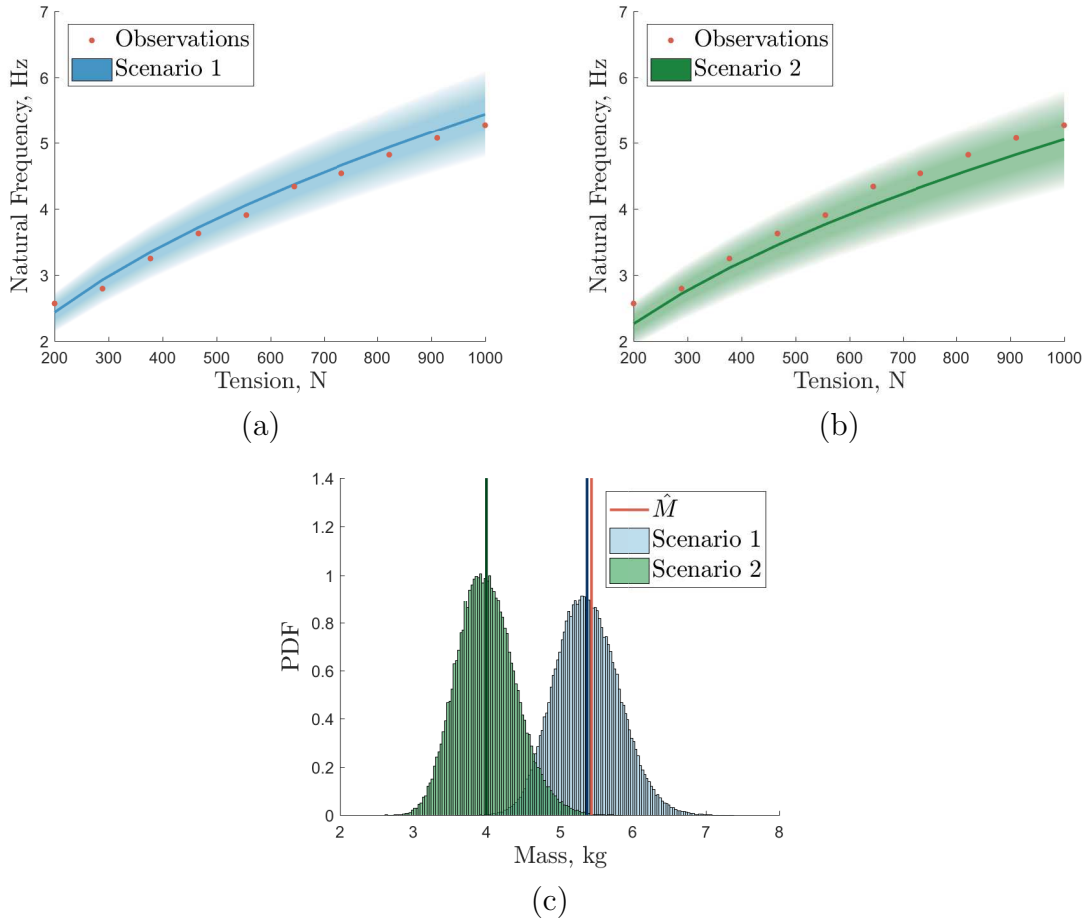


Figure 2.4: The output natural frequencies and mass distributions from the two mass, tensioned wire system scenarios. Panel (a) presents scenario 1, where the simulator and observations are from Eq. (2.2) and panel (b) scenario 2, where the simulator is from Eq. (2.1) and the observations from Eq. (2.2); the shaded regions present the output distribution. All observations have additive Gaussian noise $e \sim \mathcal{N}(0, 0.01)$. Panel (c) displays the two posterior parameter distributions for each scenario compared to the 'true' parameter $\hat{M}$.

value. This means that by not considering model discrepancy, inference of the 'true' parameter distribution will not be achievable. In addition, model form errors and bias introduced in the inferred parameter distribution will lead to problems in the output predictions; which are especially concerning for forward model-driven SHM. Figure 2.4b demonstrates the introduced problems in the output predictions, which are highlighted by a Normalised Mean Squared Error (NMSE) (for the mathematical definition of NMSE the reader is referred to Section 4.2.2) of 5.4 compared to 1.5 for scenario one. Furthermore, the model form errors in scenario two have led to an increase in variance $\approx 30\%$ from scenario one, which would lead to extra complexity

Algorithm 1 Metropolis-Hastings random walk

Set the proposal $q\left(\boldsymbol{\theta}^* \mid \boldsymbol{\theta}^{i-1}\right) = \mathcal{N}\left(\boldsymbol{\theta}^{i-1}, V\right)$
R =chol(V) ▷ Calculate the Cholesky decomposition of V
Set $\boldsymbol{\theta}^0$ ▷ Set the initial state in the Markov Chain

for $i = 1 : N$ **do**
  $\boldsymbol{\theta}^* = \boldsymbol{\theta}^{i-1} + R\varepsilon$ where $\varepsilon \sim \mathcal{N}(0, 1)$ ▷ Take a random walk
  $r = \frac{p(\boldsymbol{z} \mid \boldsymbol{\theta}^*)p(\boldsymbol{\theta}^*)}{p\left(\boldsymbol{z} \mid \boldsymbol{\theta}^{i-1}\right)p\left(\boldsymbol{\theta}^{i-1}\right)}$ ▷ Compute the ratio
  $u^i \sim \mathcal{U}(0, 1)$
  if $u^i \leq \min(1, r)$ **then**
    $\boldsymbol{\theta}^i = \boldsymbol{\theta}^*$ ▷ Accept the sample
  else
    $\boldsymbol{\theta}^i = \boldsymbol{\theta}^{i-1}$ ▷ Reject the sample
  end if
end for

in inferring decision bounds for health diagnostics in SHM.

In conclusion, model discrepancy must be considered within the calibration approach otherwise correct parameter inference is not obtainable and output predictions will suffer. This is especially problematic for forward model-driven SHM as output predictions from the simulator must be representative of those obtained in-service. Chapters 5 and 6 outline and develop methods for accounting for model discrepancy within the calibration process.

2.1.4 Multi-Level Uncertainty Integration

A significant challenge in the development of forward model-driven approaches to SHM is that the methodologies employed must not require damage state data at a full-system level, otherwise a data-driven method could be implemented that would both perform better and at a reduced procedural complexity. This constraint means that the problem of generating damage feature predictions has to be de-constructed into sub-systems for which validation is possible. This depends on the assumption that by capturing and correcting bias in the functional form of sub-system level simulators, all damage mechanisms can be modelled and validated leading to a valid full-system prediction when the sub-system simulators are combined.

Multi-level uncertainty integration offers a methodology for taking sub-system level simulators where key model forms can be validated, such as the functional relationship

when damage is introduced, and scale the uncertainties and model discrepancies through to a full-system level prediction. This means that the damage mechanisms are validated at a sub-system level, reducing the need for validation at a full-system level. This is possible if the damage mechanics can be captured at a sub-system level and appropriately scaled up. Chapter 7 outlines a potential strategy for performing multi-level uncertainty integration using a subfunction discrepancy approach.

2.1.5 Health State Decision Strategies

Health decision strategies have been well studied within the data-driven framework where a variety of machine learning methods have been successfully implemented when labelled damage state data is available, as discussed in Section 1.2.2. All of these techniques are applicable to a forward model-driven framework, with the key difference being that the labelled health state data is generated from a simulator rather than from observational data. A health decision strategy trained using a full-system simulator provides additional insight. Firstly, any classified observational data will relate to a damage state in the simulator, aiding the interpretation of the type, location and extent of damage in the structure. Furthermore, once identified the simulator can be used for prognosis, which is a significant challenge for data-driven methods. In addition any health state data collected from the structure in operation can help recalibrate and validate the simulators within the forward model-driven framework. This means that the SHM system will continue to improve over time and increase physical insight into structural behaviour under operational conditions. It is noted that any operational data obtained can also be incorporated into the training data set for the classification method as and when it becomes available.

Bayes risk classifiers are one method for making decisions about the health state of a structure [39]. The technique aims to weight known outcome probabilities of events (i.e. undamaged, de-lamination, cracks etc.) by the costs of that outcome occurring. This allows a process whereby decision bounds are formulated as a function of the likelihood of particular damage scenarios, their associated maintenance costs and the cost of structural failure. A difficulty with implementing Bayes risk for SHM is the ability to obtain the conditional probabilities of the chosen feature vector given local damage states in particular regions, i.e. the probability of a feature vector given some form of damage event. Forward model-driven SHM provides a potential solution to this challenge by using full-system simulator predictions of feature vectors

for different damage events, i.e. output predictions from the full-system simulator for the range of damage scenarios being considered.

2.2 Conclusion

Forward model-driven SHM is defined as the process of creating a full-system simulator of a particular structure for which health states can be simulated and damage sensitive features obtained. The simulated damage state features are subsequently incorporated into classification techniques where in-service data can be processed and classified online.

The biggest difficulties in generating a forward model-driven approach is that of creating a full-system simulator that accurately represents observational health state features. As a result several modelling based technologies must be investigated and developed in order to realise a robust forward model-driven approach to SHM. These methods must solve several key issues. Firstly, a clear definition of valid health state predictions must be outlined such that simulator predictions, and their statistical distributions, are close enough to those obtained in-service — this is approached in Chapter 3. If the simulator predictions are not adequately representative then this may result in confusion of the classification method. Moreover, to achieve representative predictions calibration must be employed. A consequence of forward model-driven SHM's main objective — producing representative predictions with a simulator — means that model discrepancy must be considered during the calibration process, in order to account for model form errors; this is investigated in Chapters 5 and 6. An additional complication to forward model-driven SHM is that calibration must be performed without obtaining full-system health state data. This leads to the challenge of using sub-system level data to calibrate a full-system level simulator, addressed using a multi-level uncertainty integration strategy in Chapter 7.

The forward model-driven framework proposed within this chapter highlights five areas of further development: model development and selection, damage feature selection and monitoring system design, simulator calibration (and validation), multi-level uncertainty integration and health state decision strategies. Based on the aforementioned challenges this thesis seeks to develop methodologies for approaching simulator calibration in Chapters 5 and 6 (and validation in Chapter 3) as well as developing methods for multi-level uncertainty integration in Chapter 7.

Chapter 3

Validation Metrics

Validation is a crucial part of any model generation (especially for complex simulators), without which trust in outputs for specific input domains cannot be obtained. This is especially vital in forward model-driven SHM where confidence must be obtained in order to know that the simulator predicts statistically representative outputs; otherwise the process cannot be guaranteed to appropriately detect damage. The term validation is broadly applied in many aspects of engineering with several different connotations. For clarity here validation refers to a process of quantifying the measure of fit between the simulator outputs and observational data, to ascertain the appropriateness as well as developing confidence in the simulator for its intended context of use [40].

A validation procedure requires obtaining observational data — for a forward model-driven SHM context this means collecting data from damage states. Potential solutions to this particular problem are discussed in Chapter 7. This chapter deals specifically with the challenge of validating probabilistic outputs. This is required as the objective of forward model-driven SHM is to generate statistically representative outputs, which in turn leads to the use of probabilistic UQ techniques. Consequently the chapter seeks to outline a validation strategy for probabilistic outputs (which can be applied more broadly to engineering simulators) detailing validation metrics that could be used.

3.1 Validation

In 1987 George Box famously articulated with his colleague Norman Draper that "All models are wrong, but some are useful." [41]. The statement, originally applied to statistical models, also holds true for engineering simulators. As a result the role of validation within engineering is often determining the 'usefulness' of a particular simulator for a given context or use case. The objective can be seen as more than an assessment of how 'right' the simulator is but whether the simulator is fit for purpose. Consequently, the initial starting point for a validation strategy is determining what the simulator will be used for and within what context. For forward model-driven SHM the simulator(s) are utilised for making predictions of particular damage mechanisms, with the aim of producing statistically representative damage state data to that obtained via physical observations. The context is that a simulator is deemed appropriate if the outputs lead to inferred decision bounds with an adequate rate of detection — to be determined by the SHM application. This chapter aims to outline a validation procedure and validation metrics that aid the objectiveness of decision making about adequacy of the simulator(s) for this context. However, it is noted that these decisions are often subject to the particular industrial requirements. Within the field of Verification and Validation (V&V) attempts have been made to formalise validation procedures such as the ASME V&V-20 [42], yet there is not a formalised and accepted procedure for dealing with probabilistic simulators within the community.

For clarity of terminology a *validation metric* here refers to mathematical operators that quantify the dissimilarities between predictions and observational data. A *metric*, where used on its own, refers to the mathematical distance definition; a distance $D(\cdot, \cdot)$ is a metric if it abides by four requirements [40]:

1. Non-negative: $D(x, y) \geq 0$
2. Identity of indiscernibles: $D(x, y) = 0$ if and only if $x = y$
3. Symmetric: $D(x, y) = D(y, x)$
4. Triangle inequality: $D(x, z) \leq D(x, y) + D(y, z)$

where x, y and z are three quantities — in the simplest case points. It may be

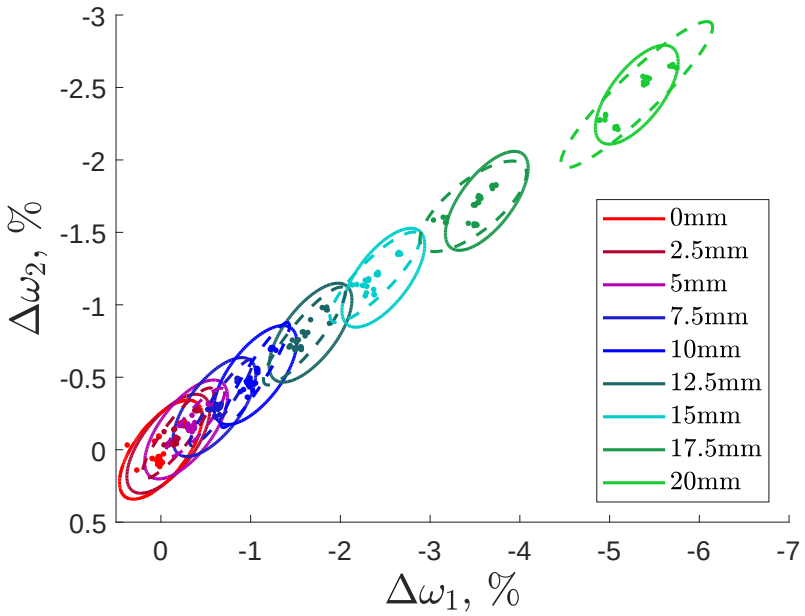


Figure 3.1: An illustration of validating statistical distributions. The example shows the simulator output distributions $(-)$ — the percentage change in the first and second natural frequencies, $\Delta\omega_1$ and $\Delta\omega_2$ respectively — for a cantilever beam with a crack of various sizes against the experimental observations $(\cdot)$ and their Gaussian representations $(--)$.

necessary for a validation metric to be a mathematical metric, the merits of this will be discussed further within the chapter.

The goal of each simulator within forward model-driven SHM, as outlined by Chapter 2, is to predict statistical distributions that capture the behaviour of observed damage state data. An illustration of the problem is demonstrated in Fig. 3.1. Rather than the traditional deterministic view, where a difference would indicate the accuracy of the simulator, the simulated responses are distributions and the observational data often a collection of measured responses. A comparison of the means or even low order statistical moments (such as variance or skewness) often will not sufficiently quantify the simulator adequacy. Scenarios could be observed where the expectations of the simulator and observations are extremely close but where there is significant mismatch in the remaining probability mass. In these cases the adequacy of a simulator may be obscured by ignoring all the available information. This leads to the conclusion that validation procedures, and more specifically the validation metrics used to quantify the appropriateness of the simulator, should involve the complete distribution and/or data point set.

The following sections outline a strategy for validating probabilistic simulators more

generally, but with a focus on the context of forward model-driven SHM. The particular tools and validation metrics that may be employed as part of a validation strategy are outlined and discussed.

3.1.1 A Validation Strategy

Any proposed validation strategy must assess how simulator adequacy will be approached. In a probabilistic setting, especially in forward model-driven SHM, the simulator's aim is to produce outputs that define or come from the same underlying distribution as observed damage state data. The definition of adequacy may depend on the degree of overlap between one damage state distribution and the next. In Fig. 3.1 when damage is below 7.5mm there is significant overlap in the distributions. If the simulator predictions were to have inflated variance or heavier tails in this region the inferred decision boundary may become more difficult to infer than if observed damage state data were used. In contrast if there is a better degree of separability, such as in Fig. 3.1 when damage is greater than 17.5mm, then this level of inadequacy may matter less. Nonetheless the goal of probabilistic simulator predictions are that it is statistically significant that the observed data plausibly came from the predicted simulator distribution.

A key statistical tool for assessing whether one distribution is not statistically similar to another is hypothesis testing. The premise of hypothesis testing is to state a statistically plausible claim (otherwise the test is irrelevant) and assess the data given that claim to decide whether it is statistically significant to reject that claim. Debate is had as to whether hypothesis testing is a useful or even a desired tool for validation. The objections to hypothesis testing come from the same ideas as Box and Draper, e.g. the statement by Oberkampf and Roy that "Any model can be proven false, given enough data" [40]. Indeed hypothesis testing as the only validation method within a strategy would be problematic, as the results do not state a measure of inadequacy or give diagnostics for simulator improvement. Another complaint is that hypothesis testing is subjective to the modeller, as they have to devise their own hypothesis (which must be testable and conceivable) as well as stating the level of statistical significance to which the hypothesis can be rejected. Undeniably this may cause problems but it would be naive to ignore the fact that many assumption are made by a modeller in the construction of a simulator. Certainly good practice, at a minimum, would be for the modeller to state all assumptions about the hypothesis test and

if possible for an external individual to perform the hypothesis test. Furthermore hypothesis testing can provide a first pass in a validation scheme. It can help decide statistically whether further interrogation is required. It is therefore the author's opinion that hypothesis testing is a useful initial tool within a wider validation strategy.

The next stage in assessing simulator validity is acquiring a quantitative assessment of the difference between simulator predicted and observed outputs. In a deterministic setting the difference (or even percentage difference) between results provides a clear and interpretable assessment of the adequacy. In a probabilistic setting several distance metrics are available for assessing the difference between distributions, these are explored in Section 3.3. These may provide more information about how inadequate the simulator is compared with hypothesis testing. At this point it is appropriate to define, in the author's opinion, a criteria for evaluating the appropriateness of validation metrics for probabilistic engineering simulators; these are:

1. It should quantify the difference between the simulator predictions and observational data.
2. It should be interpretable and aid identifying simulator improvements.
3. It should provide objective information and be consistent when applied to different probabilistic models or applications.
4. It should account for the complete form of the distributions (and not just statistical moments) - if the underlying distribution of the observational data is unknown it should have a non-parametric estimator.

The third stage of the strategy is to use visual diagnostic tools. These provide a method for determining sources of inadequacy aiding simulator improvements. This stage will often provide a high amount of information as to the source of difference between simulator and observations, but may also be more subjective.

Finally standard deterministic metrics can be implemented to assess mean (or modal) prediction validity. A deterministic approach should be taken with caution as by considering the mean (or mode) only results in discarding information from the predicted distribution.

The proposed strategy is summarised as:

- Hypothesis testing
- Quantification using probabilistic validation metrics (e.g. distance metrics)
- Visual diagnostics (e.g. witness function, Quantile-Quantile Plot (QQ-Plot))
- Deterministic validation metrics

Each layer aims to provide more detail about the sources of inadequacy and poor model performance. The follow sections aim to outline tools within this strategy before applying them to numerical examples in order to conclude about their performance.

3.2 Hypothesis Testing

Hypothesis testing, also known as significance testing, is a statistical method for deciding whether to reject or fail to reject a given hypothesis. These hypotheses are usually inferred given a one- or two-sample based problem. In the validation context described by Fig. 3.1 it will often be the case that an independent two-sample test is required. This follows as the decision about whether the simulator output is not invalid[1] will involve sets of observational samples and simulator predictions (either in known distribution form or as samples). A two-sample hypothesis test states that given two sets of finite independent and identically distributed (i.i.d.) samples, $X \sim \mathbb{P}$ and $Y \sim \mathbb{Q}$, a statistical test can be formed in order to distinguish between the null hypothesis $\mathcal{H}_0 : \mathbb{P} = \mathbb{Q}$, or an alternative hypothesis $\mathcal{H}_a : \mathbb{P} \neq \mathbb{Q}$; where $\mathbb{P}$ and $\mathbb{Q}$ are probability measures. The statistical test is formed by calculating a test statistic T and comparing this to a threshold t specified by a significance level α, where the null hypothesis is rejected if the test statistic exceeds the threshold. A hypothesis test starts with the belief that the null hypothesis is true, and works by proof of contradiction. This means that the threshold is determined by an α level for the distribution P of the test statistic T under the assumption that $\mathcal{H}_0$ is true, i.e. $P(T > t) \leq \alpha$ (where $P(T > t)$ is referred to as a probability value (p-value)).

[1] Statistically a hypothesis cannot be proved, and therefore determined completely valid.

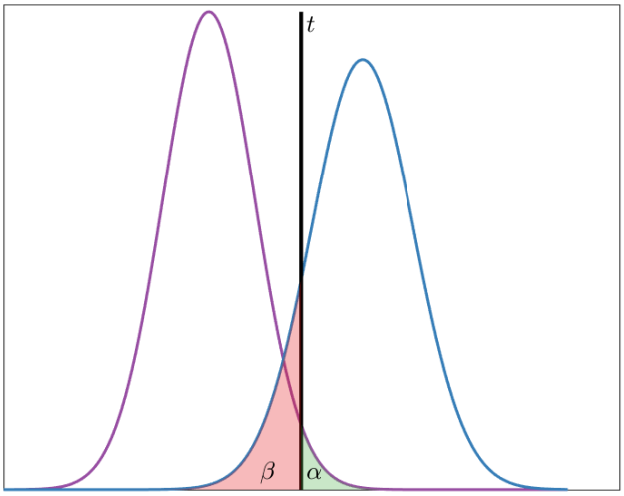


Figure 3.2: An illustration of the probability of type I error α and type II error β.

A hypothesis test by construction makes a decision based on a finite set of samples. As a consequence incorrect decisions can be made from two types of error source: type I and II. An incorrect decision based on the null $\mathcal{H}_0$ being rejected when it is actually true is defined as type I error — known as false positives — in contrast to type II error — known as false negatives — where the null is accepted when it is actually false. One minus the type II error is known as the power of the statistical test. Put another way, type I errors can be seen as rejecting a valid simulator and type II errors as not rejecting an invalid simulator. Hypothesis tests are determined as α-level tests where the value for α defines the upper bounded probability of I errors. In addition a hypothesis test can be considered consistent if it is possible for type II errors to be zero in the limit of an infinite sample size. As a result an α-level hypothesis test should aim for the probability of type II error, defined by β, to be as low as possible whilst bounding type I error at the prescribed value. Figure 3.2 provides an illustration of the two types of error. Typically within the statistical community a significance level of 5% is implemented however there is not complete consensus and for many applications a lower value should be used.

The approach shown so far defines a class of frequentist methods for testing hypotheses. In Sections 3.2.1 and 3.2.2 examples of specific frequentist significance tests that could be appropriate for validation within a forward model-driven SHM context are presented. Section 3.2.3 provides an alternative philosophical view, outlining a Bayesian approach to hypothesis testing.

3.2.1 Kolmogorov-Smirnoff Test

The Kolmogorov-Smirnoff (KS)-test, a well established hypothesis test, is constructed from a Cumulative Density Function (CDF) based test statistic, specifically the Kolmogorov distance. The Kolmogorov distance is the maximum L_1 norm between two CDFs bounded $[0, 1]$ and mathematically defined by Eq. (3.1).

$$D_K(\mathbb{P}, \mathbb{Q}) = \sup_{x \in \mathbb{R}} |F_P(x) - F_Q(x)| \tag{3.1}$$

Where sup is the supremum, the least upper bound of pointwise differences and $F_P(x)$ is a CDF for the probability measure $\mathbb{P}$ over the random variable x. Figure 3.3 illustrates an example of the distance for a set of samples (forming an Empirical Cumulative Density Function (ECDF)[2]) $\hat{F}_Q(x)$ and a known distribution $F_P(x)$ — however the distance holds if either $\mathbb{P}$ or $\mathbb{Q}$ are known or empirical. Simply the Kolmogorov distance is the largest vertical difference between the two CDFs. A strength of the Kolmogorov distance, and hence the KS-test is the ability to handle any empirical and/or known CDFs, making it a flexible non-parametric tool for validation purposes. A one sample test compares an ECDF with a CDF and two sample test, two ECDFs.

The KS-test is a hypothesis test for one-dimensional CDFs where the null hypothesis is $\mathcal{H}_0 : F_P(x) = F_Q(x)$ [43]. The Kolmogorov theorem states that as the number of samples tends to infinity, if the null hypothesis cannot be rejected, then $\sqrt{n} D_K(\mathbb{P}, \mathbb{Q})$ tends to a Kolmogorov distribution that is not dependent on the hypothesised distribution, where n are the number of samples. For a two sample test the same theorem holds, however the quantity becomes $\frac{mn}{n+m}^{1/2} D_K(\mathbb{P}, \mathbb{Q})$, where n and m are the number of points for each sample. The threshold t for a particular significance level α is often obtained using predefined tables, such as the Miller approximation table [44], for which the null hypothesis $\mathcal{H}_0$ is rejected when $D_K(\mathbb{P}, \mathbb{Q}) > t(\alpha)$. The test can also be performed by comparing whether $\alpha > p(X \mid \mathcal{H}_0)$ — the significance level against the p-value. The KS-test also has an asymptotic power $(1 - \beta)$ of 1, meaning that type II error will reach zero given an infinite sample size. However a drawback of the test is that it is often more sensitive to deviations between the distributions in the centre than the tails.

[2] An ECDF is mathematically defined as $\hat{F}_N(x) = \frac{1}{n} \sum_{i=1}^{n} \mathbb{1}(X_i \leq x)$.

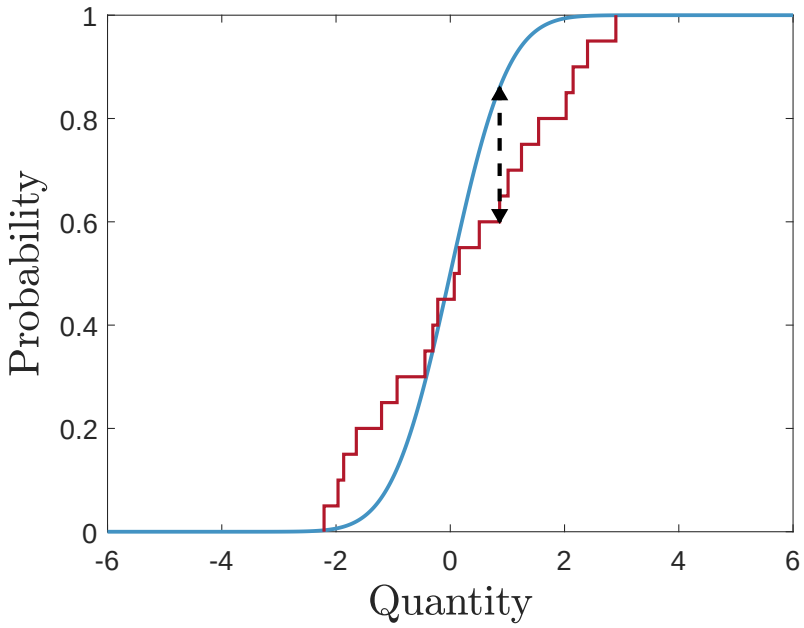


Figure 3.3: An example of the Kolmogorov distance between $\mathbb{P} = \mathcal{N}(0, 0.8^2)$ and 20 samples from $\mathbb{Q} = \mathcal{T}(5)$ where $D_K(\mathbb{P}, \mathbb{Q}) = 0.26$.

3.2.2 Maximum Mean Discrepancy Test

The Maximum Mean Discrepancy (MMD) two sample test, a relatively new technique, uses the MMD distance as a test statistic in order to distinguish between the null hypothesis $\mathcal{H}_0 : \mathbb{P} = \mathbb{Q}$ and the alternative hypothesis $\mathcal{H}_0 : \mathbb{P} \neq \mathbb{Q}$ [45]. MMD is a measure of the maximum distance between the mean embeddings of two sample sets in a Reproducing Kernel Hilbert Space (RKHS); projected using the function class $\mathcal{F}$, where the function f is called a reproducing kernel $k(\cdot, \cdot)$. The distance is defined in Eq. (3.2).

$$D_{MMD}(\mathbb{P}, \mathbb{Q}) = \sup_{f \in \mathcal{F}} |\mathbb{E}_x(f(x)) - \mathbb{E}_y(f(y))| \tag{3.2}$$

Where x and y are samples from $\mathbb{P}$ and $\mathbb{Q}$ respectively. There are several kernel types that can be chosen within the MMD metric with a popular choice being the radial basis kernel Eq. (3.3). For most kernel types there are a set of hyperparameters that need to be determined, e.g. σ for the radial basis kernel.

$$k(x, x') = \exp\left(-\frac{\|x - x'\|^2}{2\sigma^2}\right) \tag{3.3}$$

A common approach for determining these hyperparameters is to use the median pairwise distance among the joint data [46]. The choice of kernel should reflect the prior belief about the smoothness of underlying distribution and are often selected in a heuristic manner. However, Gretton et al. proposed an optimisation methodology for large sample sets whereby for a given α level the technique selects linear combinations of kernels that minimises the probability of type II errors and hence maximises the test power [47]. The method has been show to perform well in large data sets where the median heuristic start to fail, and kernel selection via selecting the kernel with the largest MMD fails. In contrast, most validation tasks will involve small sample sizes where the limited data could pose challenges to implementing this procedure.

MMD is a frequentist statistic and thus can be empirically estimated in both unbiased and biased forms, depending on whether the U-statistics or V-statistics are used to calculate the sample means. These two forms are shown in Eq. (3.4) and Eq. (3.5).

$$D^2_{MMDu}(\mathbb{P},\mathbb{Q}) = \frac{1}{m(m-1)}\sum_{i=1}^{m}\sum_{j\neq i}^{m} k(x_i,x_j) + \frac{1}{n(n-1)}\sum_{i=1}^{n}\sum_{j\neq i}^{n} k(y_i,y_j) - \frac{2}{mn}\sum_{i=1}^{m}\sum_{j=1}^{n} k(x_i,y_j) \quad (3.4)$$

$$D^2_{MMDb}(\mathbb{P},\mathbb{Q}) = \frac{1}{m^2}\sum_{i,j=1}^{m} k(x_i,x_j) + \frac{1}{n^2}\sum_{i,j=1}^{n} k(y_i,y_j) - \frac{2}{mn}\sum_{i,j=1}^{m,n} k(x_i,y_j) \quad (3.5)$$

Where m and n are the number of points in the samples X and Y respectively. These two forms of the statistic will both be zero when $\mathbb{P} = \mathbb{Q}$ and large when the distributions are far apart.

The hypothesis test uses the quantity $mD^2_{MMD}(\mathbb{P},\mathbb{Q})$ (either in biased or unbiased form) in comparison to the threshold $t(\alpha)$ in order to determine whether the null hypothesis $\mathcal{H}_0 : \mathbb{P} = \mathbb{Q}$ can be rejected, i.e. $mD^2_{MMD}(\mathbb{P},\mathbb{Q}) > t(\alpha)$, where α is an upper bound on the probability of type I errors. The MMD two sample test is also shown to be consistent [45]. The threshold $t(\alpha)$ is calculated via a bootstrap approach where a data-dependent threshold is estimated from calculating the test statistic from random permutations of the samples and finding the $(1-\alpha)$th quantile

[45, 48].

The approach has been implemented within the machine learning community as a model criticism technique [48, 49]. In addition to the kernel (MMD) two sample test, one sample test formulations have been created based on the Kernel Stein Discrepancy (KSD) [50, 51] which may also be appropriate for validation.

3.2.3 Bayesian Hypothesis Test

Bayesian statistics offer a different view of hypothesis testing (see Appendix A.1 for details on Bayes' theorem). This approach means that rather than using the probability of the data given a hypothesis i.e. $p\left(x \mid \mathcal{H}_1\right)$, the posterior $p\left(\mathcal{H}_1 \mid x\right)$ — the probability of the hypothesis given the data — is utilised, constructed as shown in Eq. (3.6).

$$p\left(\mathcal{H}_1 \mid x\right) = \frac{p\left(x \mid \mathcal{H}_1\right) p\left(\mathcal{H}_1\right)}{p\left(x\right)} \tag{3.6}$$

This can be compared to a second hypothesis $\mathcal{H}_2$ by $p\left(\mathcal{H}_2 \mid x\right) = 1 - p\left(\mathcal{H}_1 \mid x\right)$ as the evidence of our data $p\left(x\right)$ incorporates the possibility that each of the stated hypothesis being considered could be true, $p\left(x\right) = p\left(x \mid \mathcal{H}_1\right) p\left(\mathcal{H}_1\right) + p\left(x \mid \mathcal{H}_2\right) p\left(\mathcal{H}_2\right)$. By considering the ratio of the posteriors (also referred to as the posterior odds) Eq. (3.7) can be defined.

$$\frac{p\left(\mathcal{H}_1 \mid x\right)}{p\left(\mathcal{H}_2 \mid x\right)} = \frac{p\left(x \mid \mathcal{H}_1\right) p\left(\mathcal{H}_1\right)}{p\left(x \mid \mathcal{H}_2\right) p\left(\mathcal{H}_2\right)} \tag{3.7}$$

From Eq. (3.7) the Bayes factor, the evidence of data x for $\mathcal{H}_1$ over $\mathcal{H}_2$ can be formulated as in Eq. (3.8). As stated, for a continuous distribution this is the ratio of marginal likelihoods i.e. the ratio of our data given all possible parameters of the model and hypotheses.

$$BF = \frac{p\left(x \mid \mathcal{H}_1\right)}{p\left(x \mid \mathcal{H}_2\right)} = \frac{\int p\left(x \mid \theta, \mathcal{H}_1\right) p\left(\theta \mid \mathcal{H}_1\right) \mathrm{d}\theta}{\int p\left(x \mid \theta, \mathcal{H}_2\right) p\left(\theta \mid \mathcal{H}_2\right) \mathrm{d}\theta} \tag{3.8}$$

When priors for the hypotheses are too difficult to elicit, or not known, Bayes factor can be implemented as a test of hypothesis instead, for example Sankararaman and

Mahadevan use Bayes factor within an engineering validation setting [52]. However, Bayes factor may not strictly be justified as a clear hypothesis test given the exclusion of the prior odds $p(\mathcal{H}_1)/p(\mathcal{H}_2)$. A more reasoned interpretation is that Bayes factor is a measure how the data has changed the odds of $\mathcal{H}_1$ relative to $\mathcal{H}_2$ [53]. It is noted that Bayes factor is similar to the ratio of BIC, which is a measure of $\log p(x \mid \mathcal{M}_i)$ where a model $\mathcal{M}$ can be thought of as a hypothesis $\mathcal{H}$ [33].

Model Reliability metric

The model reliability metric is one form of hypothesis used to create a validation metric [54, 55]. This validation metric assesses whether the simulator output y and observation data $z \sim \mathcal{N}(\mu_z, \sigma_n^2)$ are less than a given tolerance λ. The probability of this hypothesis, that the simulator is valid $\mathcal{H}_V$, known as the model reliability metric, can be constructed from Eq. (3.9).

$$p(\mathcal{H}_V \mid z) = p(|y - z| < \lambda) \tag{3.9}$$

However, if σ_n^2 and y are considered deterministic then Li and Mahadevan state the metric becomes an integral of a univariate Gaussian distribution over the tolerance bounds, as shown in Eq. (3.10) [55].

$$p(\mathcal{H}_V \mid z) = \int_{-\lambda}^{\lambda} \frac{1}{\sqrt{2\pi\sigma_n^2}} \exp\left(\frac{\varepsilon - (y - z)^2}{2\sigma_n^2}\right) \mathrm{d}\varepsilon \tag{3.10}$$

Where ε is a dummy variable. This essentially is not a Bayesian treatment of the problem and instead is a probability measure from a Gaussian likelihood. When the simulator output is considered stochastic, due to parameter $\boldsymbol{\theta}$ uncertainties, the metric becomes the marginalisation $p(\mathcal{H}_V \mid z) = \int p(\mathcal{H}_V \mid z, \boldsymbol{\theta})\eta(\boldsymbol{\theta})\mathrm{d}\boldsymbol{\theta}$ (where $\eta(\boldsymbol{\theta})$ is a simulator). In a multivariate setting the metric becomes the probability that the Mahalanobis distance $D_M(\boldsymbol{y}, z_i) = \sqrt{(\boldsymbol{y} - z_i)^\intercal \Sigma_y^{-1} (\boldsymbol{y} - z_i)}$ (where Σ_y is the covariance matrix of $\boldsymbol{y}$) is less than the normalised tolerance $\lambda_M = \sqrt{\lambda^\intercal \Sigma_y^{-1} \lambda}$; $p(\mathcal{H}_V \mid z_i) = p(D_M(\boldsymbol{y}, z_i) < \lambda_M \mid z_i)$. Essentially the metric is the probability that the normalised principle component is less than a given tolerance. This is a limited metric as it only considers lower moments of the simulator distribution, and even then only in its principle component. It is therefore better to either consider a

statistical distance between distributions or to calculate the Bayes factor and perform a Bayesian hypothesis test in a formal sense.

3.3 Distance Metrics

Distance metrics are commonly used in a deterministic setting, as they provide a clear and interpretable method of validating a simulator. When considering probabilistic outputs, namely comparing two distributions as in Fig. 3.1, there are several distances/divergences that could be employed. The following sections discuss two families of probabilistic distance metrics: f-divergences and Integral Probability Metric (IPM)s.

3.3.1 f-Divergences

The class of distances/divergences that depend on a ratio between probability measures are known as *Csiszár's ϕ-divergences* or f-divergences. These measures are of the form defined in Eq. (3.11).

$$D_\phi\left(\mathbb{P}, \mathbb{Q}\right) = \int_M \phi\left(\frac{d\mathbb{P}}{d\mathbb{Q}}\right) \mathrm{d}\mathbb{P} \tag{3.11}$$

Where M is a measurable space and ϕ is a convex function. Equation (3.11) holds when $\mathbb{P}$ is absolutely continuous with respect to $\mathbb{Q}$ and $-\infty$ otherwise. Different forms of the f-divergence depend on the choice of function ϕ with notable cases being the Kullback-Leibler (KL) divergence, $\phi(t) = t\log(t)$, Hellinger distance, $\phi(t) = \left(\sqrt{t}-1\right)^2$, and total variation distance, $\phi(t) = |t-1|$. This family of divergence measures is widely used throughout information theory and machine learning.

Kullback-Leibler Divergence

The KL-divergence is the most widely used f-divergence and has many applications. A notable example is in performing variational inference as it is a natural formulation of the ratio between two likelihood functions [56]. The KL-divergence of probability

measures $\mathbb{P}$ and $\mathbb{Q}$ is shown in Eq. (3.12).

$$D_{KL}\left(\mathbb{P},\mathbb{Q}\right) = KL(\mathbb{P}||\mathbb{Q}) = \int p(x)\log\left(\frac{p(x)}{q(x)}\right)\mathrm{d}x \tag{3.12}$$

Where $p(x)$ and $q(x)$ are probability distributions of the random variable x. The KL-divergence is a measure of relative entropy [57] taking the units nats, or bits depending on the base of the logarithm, exponential or two respectively. The divergence informs of the average number of extra nats (or bits) required to encode the data given that the distribution $\mathbb{Q}$ is used to model the 'true' distribution $\mathbb{P}$. This can be thought of as how well $\mathbb{Q}$ approximates $\mathbb{P}$. The KL-divergence can be difficult to estimate and often proves challenging when the dimension size of samples increases (i.e. in the instants where d increases when $M = \mathbb{R}^d$). On the other hand the divergence can be practical to compute between low-dimensional probability density functions and therefore is useful when the observational density function is known or can be accurately approximated.

The KL-divergence is not a metric as it does not meet two of the four requirements: it is neither symmetric nor obeys the triangle inequality. A smoothed and symmetrised form of the KL-divergence is the Jenson-Shannon divergence [58], which by taking the square root becomes a metric, known as the Jenson-Shannon distance defined in Eq. (3.13).

$$D_{JSD}\left(\mathbb{P},\mathbb{Q}\right) = \sqrt{\frac{1}{2}D_{KL}\left(\mathbb{P},\mathbb{M}\right) + \frac{1}{2}D_{KL}\left(\mathbb{Q},\mathbb{M}\right)} \tag{3.13}$$

Where $\mathbb{M} = \frac{1}{2}\left(\mathbb{P}+\mathbb{Q}\right)$ and is the midpoint. This will always produce a finite result, unlike the KL-divergence as $\mathbb{P}$ and $\mathbb{Q}$ are always absolutely continuous with respect to $\mathbb{M}$. The computational overheads of the Jenson-Shannon distance are high due to the mixture distribution $\mathbb{M}$, which becomes prohibitive in high dimensional data. In addition it is less sensitive to scenarios when distribution $\mathbb{Q}$ contains sample values that are impossible in $\mathbb{P}$, unlike the KL-divergence.

Empirical estimation of the KL-divergence in a non-parametric manner for continuous distributions can be approximated using several approaches [59, 60]. Here a non-parametric estimation method based on data-dependent partitions is used; which has been shown to be strongly consistent [59]. For the unidimensional case, assume

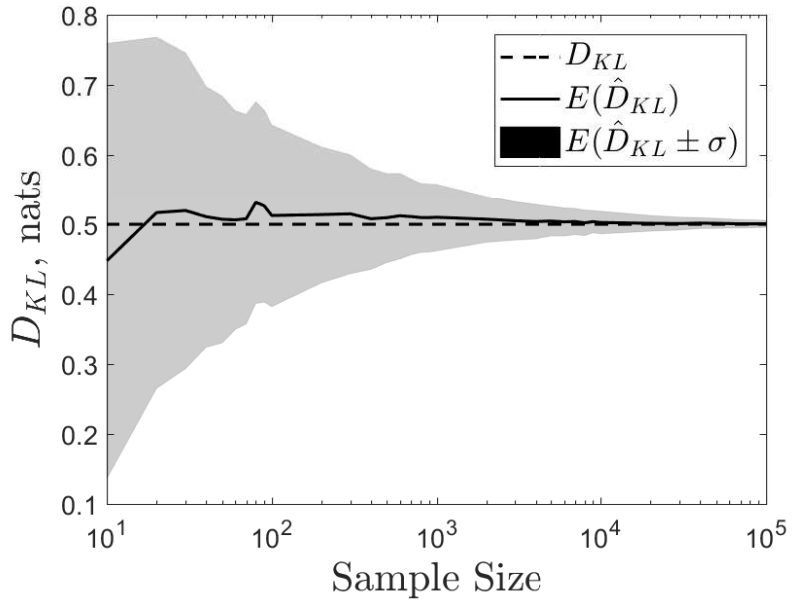


Figure 3.4: Estimation of KL-divergence using data-dependent partitions where $\mathbb{P} \sim \mathcal{N}(0,1)$ and $\mathbb{Q} \sim \mathcal{N}(1,1)$. $D_{KL}(\mathbb{P},\mathbb{Q}) = 0.5$.

i.i.d. samples from probability measures $\mathbb{P}$ and $\mathbb{Q}$; $X = \{x_1, x_2, \ldots x_m\}$ and $Y = \{y_1, y_2, \ldots y_n\}$. The algorithm orders Y so that $Y_{(1)} \leq Y_{(2)} \leq \ldots Y_{(n)}$, where $Y_{(i)}$ refers to the ith index of Y. A partition of empirically equivalent segments divides Y, called l_n spacings, as defined in Eq. (3.14), with l_n points in each interval (expect possibly the final one).

$$I^n = \{(-\infty, Y_{(l_n)}], (Y_{(l_n)}, Y_{(2l_n)}], \ldots, (Y_{(l_n(T_n-1))}, +\infty)\} \tag{3.14}$$

Where brackets have interval notation meaning, $l_n \leq n$ and $T_n = \lfloor n/l_n \rfloor$. The empirical estimate of the KL-divergence can then be calculated from Eq. (3.15).

$$\hat{D}_{KL}(\mathbb{P},\mathbb{Q}) = \sum_{i=1}^{T_n} \mathbb{P}_m(I_i^n) \log \frac{\mathbb{P}_m(I_i^n)}{\mathbb{Q}_n(I_i^n)} \tag{3.15}$$

Where $\mathbb{P}_m$ and $\mathbb{Q}_m$ are empirical probability measures. This can easily be adapted to multidimensional data. As the number of samples and partitions increase $\hat{D}_{KL}(\mathbb{P},\mathbb{Q})$ approaches $D_{KL}(\mathbb{P},\mathbb{Q})$ [59].

Figure 3.4 presents a convergence study of the empirical estimator for unidimensional samples drawn from two Gaussian distributions, $\mathbb{P} \sim \mathcal{N}(0,1)$ and $\mathbb{Q} \sim \mathcal{N}(1,1)$. 500 repeats were performed at each sample size in order to demonstrate the variance of the estimator. It is clearly presented that although the estimator will converge, this can be slow and requires a large sample size. In most engineering applications it is often not possible to obtain even hundreds of samples at each input indicating a drawback with the estimator.

Hellinger Distance

The Hellinger distance is analogous to the Euclidean distance for probability measures as it is an L_2 norm, defined in Eq. (3.16).

$$D_H(\mathbb{P},\mathbb{Q}) = \sqrt{\frac{1}{2}\int \left(\sqrt{p(x)} - \sqrt{q(x)}\right)^2 \mathrm{d}x} \tag{3.16}$$

Hellinger distance is a metric meeting all four requirements as well as having the property that $D_H(\mathbb{P},\mathbb{Q}) \leq 1$. This provides an intuitive interpretation of the distance where values close to zero mean very similar probability measures and a distance close to one indicates very dissimilar probability measures.

Total Variation

Total variation distance is the L_1-norm equivalent to the Hellinger distance and is defined in Eq. (3.17).

$$D_{TV}(\mathbb{P},\mathbb{Q}) = \sqrt{\frac{1}{2}\int |p(x) - q(x)| \mathrm{d}x} \tag{3.17}$$

This is the only distance measure that can be classed as both an f-divergence and IPM (discussed in Section 3.3.2) [61]. In IPM form, total variation is written as Eq. (3.18).

$$D_{TV}^2(\mathbb{P},\mathbb{Q}) = \sup_{||f||_\infty \leq 1} |p(x) - q(x)| \tag{3.18}$$

Total variation distance, like the Hellinger distance, takes values in $[0, 1]$ aiding interpretability and objectivity across applications.

3.3.2 Integral Probability Metrics

IPMs differ from f-divergences as they depend on the difference rather than ratio of probability measures. These measures are defined as in Eq. (3.19).

$$D_{\mathcal{F}}(\mathbb{P}, \mathbb{Q}) = \sup_{f \in \mathcal{F}} \left| \int_M f \mathrm{d}\mathbb{P} - \int_M f \mathrm{d}\mathbb{Q} \right| \tag{3.19}$$

Where $\mathcal{F}$ is a class of functions on M. The choice of $\mathcal{F}$ leads to various IPMs, such as the total variation distance where $\mathcal{F} = \{f : \|f\|_\infty \leq 1\}$, the Kolmogorov distance where $\mathcal{F} = \{\mathbb{1}_{(-\infty,t]} : t \in \mathbb{R}_d\}$, MMD where $\mathcal{F} = \{f : \|f\|_{\mathcal{H}} \leq 1\}$ (i.e. all f that are RKHS, $\mathcal{H}$) and the Wasserstein distance where $\mathcal{F} = \{f : ||f||_L \leq 1\}$ where L here refers to Lipschitz functions.

Kolmogorov Distance

The Kolmogorov distance is closely related to the total variation distance, involving CDFs as stated in Eq. (3.1). If the probability function is non-decreasing then total variation will provide the same solution as the Kolmogorov distance. Furthermore total variation is an upper bound on the Kolmogorov distance i.e. $D_K(\mathbb{P}, \mathbb{Q}) \leq D_{TV}(\mathbb{P}, \mathbb{Q})$.

Maximum Mean Discrepancy Distance

As stated in Eq. (3.2), MMD is the difference between mean embeddings in a RKHS of two finite sample sets. MMD is a non-parametric technique meaning that the form of the distribution does not need to be known before estimation.

Area Metric

The area metric, proposed by Ferson et al. [62], is a popular validation metric in engineering for assessing the difference between two distributions [40, 63–65]. The

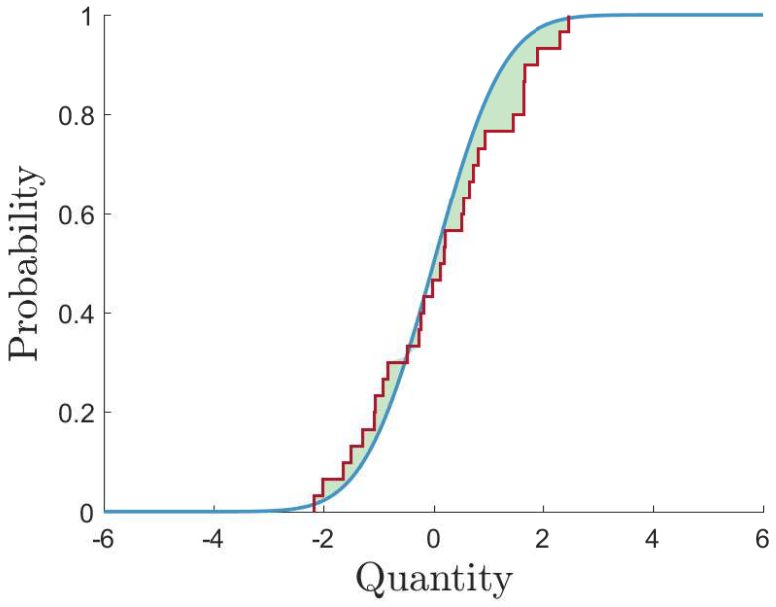


Figure 3.5: An example of the area metric (the shaded region) between $\mathbb{P} = \mathcal{N}(0, 0.8^2)$ and 20 samples from $\mathbb{Q} = \mathcal{T}(5)$ where $D_{Area}(\mathbb{P}, \mathbb{Q}) = 0.64$.

area metric is the area of the L_1-norm between two CDFs, defined by Eq. (3.20) and illustrated in Fig. 3.5.

$$D_{Area}(\mathbb{P}, \mathbb{Q}) = \int |F_P(x) - F_Q(x)| \mathrm{d}x \tag{3.20}$$

The metric is also the distance between quantile functions (inverse CDF) i.e. $\int |F_P^{-1}(p) - F_Q^{-1}(p)| dp$ where p is a probability. This means that the metric is part of the Wasserstein (or Kantorovich) distances. The metric is part of a family of metrics, known as the L_p metrics, where the L_p-norm is taken rather than L_1.

Oberkampf and Roy state that a significant merit of the area metric is that the units are that of the quantity in question, i.e. if the random variable X were an observation of stress in MPa then the area metric too is in MPa, since probability is dimensionless [40]. The distance therefore scales with the units of observed quantity.

3.4 Maximum Mean Discrepancy Witness Function

MMD, defined in Eq. (3.2), provides an additional benefit in that the kernel embedding can be applied over a variable t, in order to visualise the behaviour of the RKHS embeddings, producing the witness function, f^*. An empirical estimation of the witness function, outlined in Eq. (3.21), can be formed to provide a method for visually determining the dissimilarities between two distributions.

$$f^*(t) \propto \frac{1}{m}\sum_{i=1}^{m} k\left(x_i, t\right) - \frac{1}{n}\sum_{i=1}^{n} k\left(y_i, t\right) \tag{3.21}$$

The witness function intuitively is zero where the two distributions are the same, positive when $\mathbb{P}$ is larger and negative when $\mathbb{Q}$ is greater, as far as the smoothness constraint allows. The example in Fig. 3.6 demonstrates the information gained from calculating the witness function. A radial basis kernel is used with $\sigma = 0.85$.

The witness function can be implemented as a validation tool, where the differences help diagnose model inadequacies. For example, if in Fig. 3.6 X are simulator predictions and Y observations, it can be easily identified that more probability mass is located around zero from the sample set Y than is modelled by X; this is indicated by negative values in the witness function. In addition, X has more probability mass in both tails, indicated by the positive values in the witness function. A near symmetric witness function informs that the mean predictions are very similar. The witness function in this example would diagnose a conservative simulator output, where a distribution with a steeper probability mass decay from the mode would improve the prediction. In this one dimensional case this information may appear obvious, however this will not always be the case in more complex and bespoke distributions. Furthermore, in higher dimensional spaces it becomes challenging to compare two Probability Density Function (PDF)s and a witness function will provide a low dimensional interpretable diagnostic.

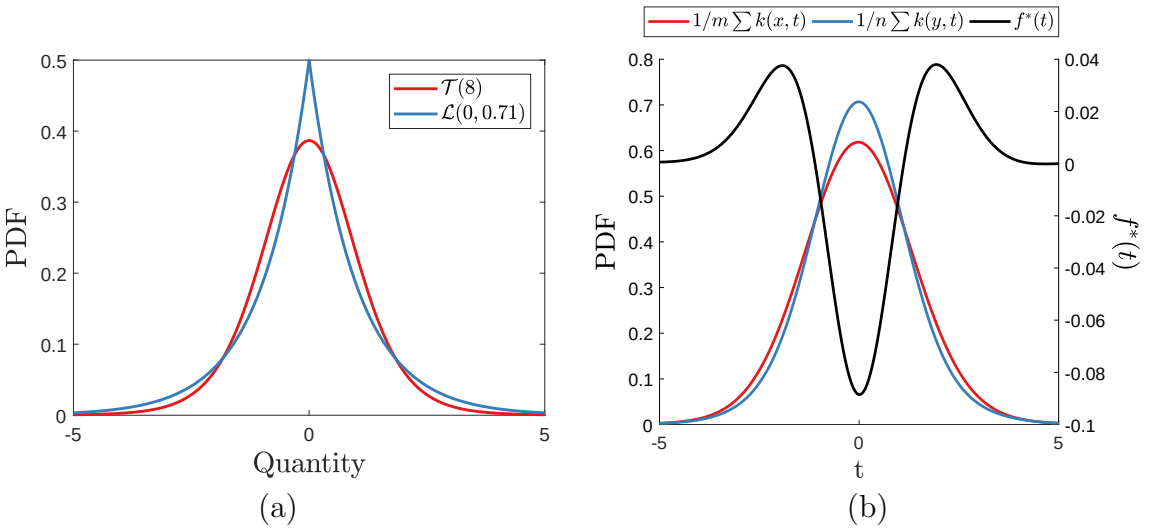


Figure 3.6: An example of a witness function between 10000 samples from $X \sim \mathcal{T}(8)$ and $Y \sim \mathcal{L}(0, 0.71)$, $D_{MMDu} = D_{MMDb} = 0.11$. Where $\mathcal{L}(\cdot, \cdot)$ and $\mathcal{T}(\cdot)$ are Laplace and Student's t distributions. Panel (a) are the PDFs of the distributions from which the finite samples are drawn and panel (b) are the mean kernel embeddings of the two samples and the witness function over a space t.

3.5 Numerical Examples

The frequentist hypothesis tests outlined in Sections 3.2.1 and 3.2.2 both rely on the sensitivity of the test statistic to changes in the distribution to determine statistical significance. This means that by assessing the distance metrics that form these test statistics, the effectiveness of the hypothesis test can be inferred. Additionally, Bayesian hypothesis tests make most practical sense when realistic hypotheses are formed. For these reasons hypothesis tests are not compared in the following numerical examples, but will be discussed and applied throughout the thesis. Instead the following numerical examples seek to assess the performance of the outlined distance/divergence measures.

In order to compare the statistical distances specified in Section 3.3 several numerical examples are considered. Continuous distributions with known mathematical forms are studied in order to analyse the behaviours of the distances in difference contexts. Practically most real scenarios will involve both sampled or one known distribution, meaning that approximators, such as a Kernel Density Estimate (KDE) or other non-parametric formulations may be used. In order to keep comparisons between distances consistent, numerical integration is implemented to calculate each distance. It is noted however that for certain known distributions it is possible to solve some

distances/divergences in closed form.

The first example seeks to determine how the distances are affected by lower order moments, specifically in the context of a Gaussian distribution. Figure 3.7 shows the distance between $\mathbb{P} \sim \mathcal{N}(0,1)$ and $\mathbb{Q} \sim \mathcal{N}(\mu_x, \sigma_x^2)$. The first case is when the mean μ_x is varied and the variance σ_x^2 is fixed, the second case considers the mean μ_x fixed and the standard deviation σ_x variable.

For the first case, where a change in mean is considered, the KL-divergence becomes symmetric (i.e. $D_{KL}(\mathbb{P},\mathbb{Q}) = D_{KL}(\mathbb{Q},\mathbb{P})$) and rapidly increases indicating sensitivity to change in the mean. On the other hand, the KL-divergence is slow to increase initially and may struggle to detect small variations in the mean. The convex shape of the KL-divergences demonstrates why it is widely used for optimisation purposes. In contrast the area metric tracks with the distance between the two distribution means i.e. when $\mu_x = 2$, $D_{Area}(\mathbb{P},\mathbb{Q}) = 2$. Comparing the distance metrics bounded [0 1] — the Hellinger, total variation and Kolmogorov distances — illustrates that total variation distance is the most sensitive to the change in mean, followed by Kolmogorov and Hellinger distances. With the knowledge that these have an upper bound of 1, the distances become quite far relatively quickly, i.e. when $\mu_x = 2$ total variation is 0.83 compared with 0.68 and 0.62 for the Kolmogorov and the Hellinger distance. For this scenario the distances can be interpreted as far away and would lead to an acknowledgement of significant inadequacy in the relationship between the simulator and observations. It is argued that these distances give a better indication of the relative difference between the distributions providing an objective comparison when compared with the KL-divergence and area metric. The MMD distances do not have an upper bound but track relatively consistently with both the Kolmogorov distance and Hellinger distances. It is noted that the MMD's non-parametric, sample based, approximation of the distributions leads to oscillations in the metrics. Additionally, both bias and unbiased results are very similar and become less sensitive to changes in the mean ≥ 4 and ≤ -4 when compared with the Kolmogorov and Hellinger distances.

The second scenario, when the variance is varied, demonstrates the asymmetric nature of the KL-divergence where more nats of information are required in order to encode $\mathbb{Q}$ when $\mathbb{P}$ is the model distribution than in the opposing case. This is because when the proposed model distribution has little or no probability mass in areas where the target distribution is expected to have probability mass, more information would required to replicate the target distribution. However, when there

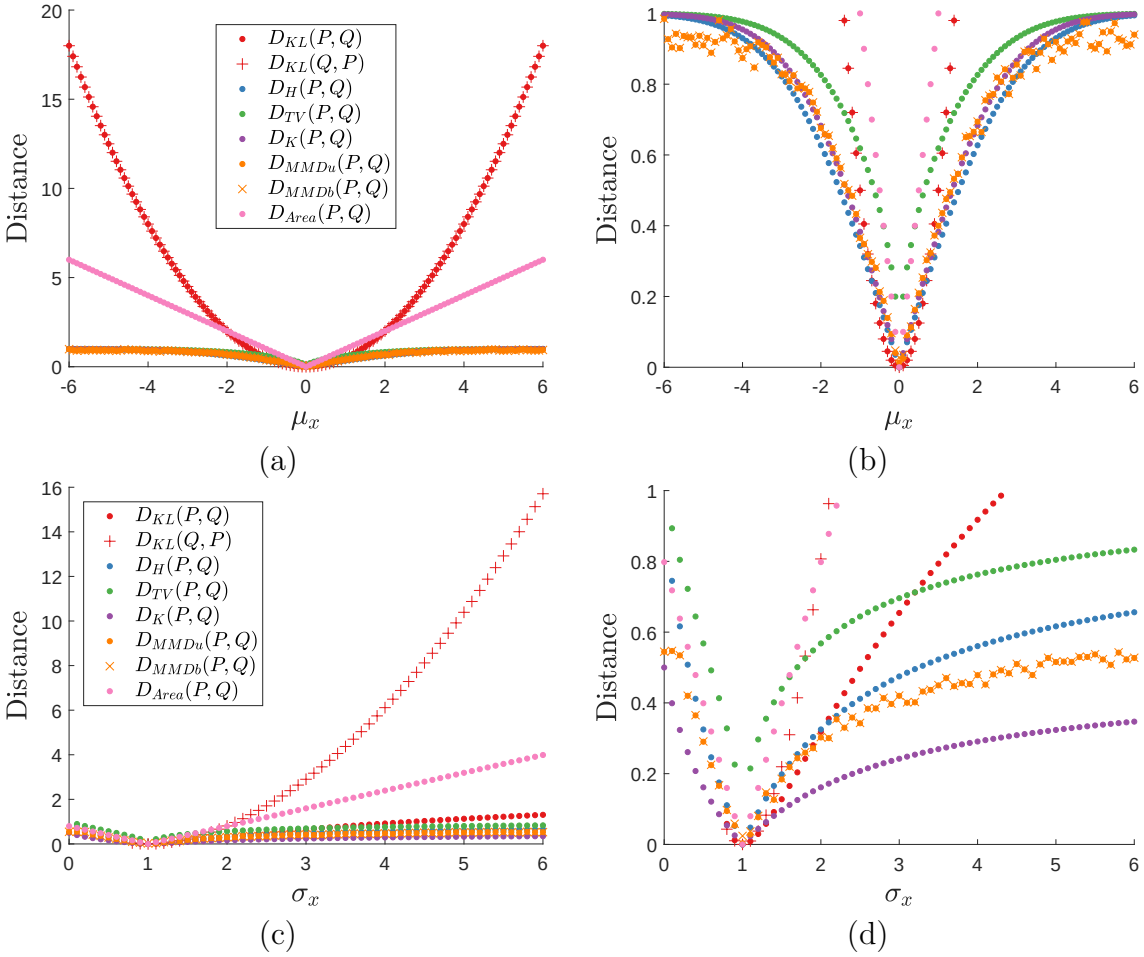


Figure 3.7: A comparison of probabilistic distances/divergences for two Gaussian distributions, $\mathbb{P} \sim \mathcal{N}(0,1)$ and $\mathbb{Q} \sim \mathcal{N}(\mu_x, \sigma_x^2)$. Panel (a) and (b) demonstrate the distances/divergences when the mean μ_x is varied from [-6 6] with a fixed variance $\sigma_x^2 = 1$. Panels (c) and (d) present the distances/divergences when the standard deviation σ_x is varied from [0 6] with a fixed mean $\mu_x = 0$. Panels (b) and (d) show the distances with y-axis limits [0 1]. It is noted that the KL-divergence units are nats and the area metric is in the units of x (in this case non-dimensional), although all distances are plot on the same axes for visualisation purposes. The MMD distance is calculated from 2000 samples and all other distances from numerical integration over the range [-30 30] in 0.01 steps.

is a broad spread of probability mass in the model distribution, covering all areas of high probability mass in the target distribution, less information is needed to encode the target distribution. This means that the KL-divergence will often favour conservative model distributions, useful for a validation setting. Unfortunately the units of the KL-divergence are difficult to intuitively interpret. The area metric linearly scales with a change in variance implying it is less sensitive to this change than the other distances/divergences. Nonetheless the area metric is valuable as the units are the same as the quantity of interest. Furthermore, the area metric appears almost symmetric about the variance of $\mathbb{P}$, suggesting the area metric suffers to differentiate between under- and over-estimations of the variance; an unhelpful property in validation. In comparison total variation, Hellinger and Kolmogorov distances appear more sensitive to underestimation of the variance, indicated by a steeper gradient of distances below a standard deviation of 1. In conjunction with the previous findings total variation is more sensitive to changes in the standard deviation than the Hellinger or Kolmogorov distances. Here the Kolmogorov distance becomes less sensitive than the Hellinger distance, this is due to the fact that the Kolmogorov distance is less sensitive to changes in the tails, compared to difference in the central probability mass. Again both MMD distances track in a similar manner to the Hellinger distance.

The next examples presented in Tables 3.1 and 3.2 compare the statistical distances for different forms of distribution. The first two examples compare standard Gaussian and Laplace distributions (with the same mean and variance) as well as standard Gaussian and Student's t distributions, where small dissimilarities are visually shown in Fig. 3.8. For these two examples the KL-divergences (in both directions) indicate that relatively small amounts of information are required to encode the 'true' distribution, from the low KL-divergences given the log ratio relationship. The Kolmogorov distance shows very small distances, expected given its insensitivity to differences away from the central probability mass. The MMD distances, both biased and unbiased, produce comparable results calculating larger distances for the Laplace than the Student's t distributions. The Hellinger and total variation distance also evidence that the standard Gaussian is closer to the Student's t distribution than the Laplace distribution, but by a relatively smaller amount. Again the Hellinger distance produces smaller distances than total variance; this is an expected result given that total variation is an upper bound to the Hellinger distance. The two area metrics for these examples are the same. This demonstrates a failure to capture the knowledge that a Student's t is expected to be closer to the standard normal than a

$\mathbb{P}$	$\mathbb{Q}$	$D_{KL}(\mathbb{P},\mathbb{Q})$	$D_{KL}(\mathbb{Q},\mathbb{P})$	$D_{H}(\mathbb{P},\mathbb{Q})$	$D_{TV}(\mathbb{P},\mathbb{Q})$
$\mathcal{N}(0,1)$	$\mathcal{L}(0,0.71)$	0.07	0.23	0.16	0.34
$\mathcal{N}(0,1)$	$\mathcal{T}(5)$	0.03	0.12	0.11	0.25
$\mathcal{G}(2,1)$	$\mathcal{N}(1,1)$	-	∞	0.38	0.50
$\mathcal{U}(-4,4)$	$\mathcal{N}(0,1)$	-	∞	0.46	0.70

Table 3.1: Examples of f-divergences for different distributions. Numerically integrated over the range [-30 30] in 0.01 steps. KL-divergences are in nats.

$\mathbb{P}$	$\mathbb{Q}$	$D_{K}(\mathbb{P},\mathbb{Q})$	$D_{MMDu}(\mathbb{P},\mathbb{Q})$	$D_{MMDb}(\mathbb{P},\mathbb{Q})$	$D_{Area}(\mathbb{P},\mathbb{Q})$
$\mathcal{N}(0,1)$	$\mathcal{L}(0,0.71)$	0.06	0.12	0.12	0.15
$\mathcal{N}(0,1)$	$\mathcal{T}(5)$	0.03	0.04	0.05	0.15
$\mathcal{G}(2,1)$	$\mathcal{N}(1,1)$	0.25	0.26	0.26	1.00
$\mathcal{U}(-4,4)$	$\mathcal{N}(0,1)$	0.25	0.44	0.44	1.20

Table 3.2: Examples of IPM distances for different distributions. Numerically integrated over the range [-30 30] in 0.01 steps apart from the MMD distances which are estimated from 2000 samples.

Laplace distribution.

The KL-divergence for the next two examples, a comparison of Gamma and Gaussian distributions and uniform and Gaussian distributions, presents issues with using numerical integration, but provides informative results. As the Gamma distribution contains no probability mass below zero it is logical for it to be impossible for a Gaussian distribution, that has symmetric probability mass over $-\infty$ to ∞ range, to ever be be able to replicate the Gamma distribution, given any amount of additional information. In contrast a Gamma distribution would require an infinite amount of additional information below zero to replicate the Gaussian distribution. The KL-divergence, calculated in this manner, is extremely informative in diagnosing these issues. Similar problems also exist in the comparison of a uniform and Gaussian distributions, where a uniform distribution contains no probability mass outside of its range. The Kolmogorov distances for these examples are the same, contributing more evidence of issues with the distance when differences are outside the central probability mass. Moreover, the total variation, Hellinger and MMD distances, including the area metric, all evidence that the uniform and Gaussian distribution distances are further than the Gamma and Gaussian distribution. Once more the total variation is more sensitive than the Hellinger distance to difference in the distributions with the MMD distances being most similar to the Hellinger distance.

The results from empirical numerical observations indicate the strength and weak-

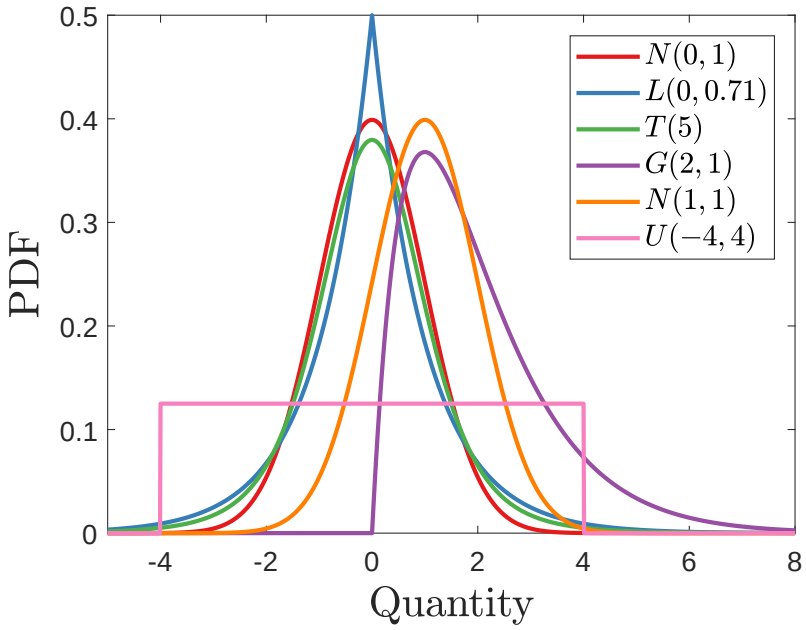


Figure 3.8: Distributions used in the comparison of distance/divergences.

nesses of the distances/divergences considered. It can be summarised that the KL-divergence becomes very sensitive in scenarios where large amounts of extra information are required to replicate the 'true' distribution, and its convex nature makes it ideal for optimisation settings. This makes the divergence useful for scenarios when the question of whether to obtain more observations or simulator runs to solve issues of inadequacy are asked. In the event of a particularly large (or even ∞) divergence it can be inferred that the proposed model distribution form is not appropriate. The major drawback of the KL-divergence is that outside of these extremes it is not easily interpretable. The Kolmogorov distance is flawed as a general distribution validation metric for the aforementioned reasons, and it is not recommended as the sole qualification of the distance between distributions. The total variation and Hellinger distances can arguably be seen are more interpretable and objective in comparing two distributions given that 0 indicates they are the same and 1 that the distributions are as far as possible. Total variation is more sensitive than Hellinger and may be more sensitive than required for engineer validation applications. The Hellinger distance, in the author's opinion, seems more intuitive given the results in Table 3.1. Furthermore, the MMD distances tend to provide similar distances to the Hellinger and may be practical in a variety of settings due to its non-parametric formulation. However for small sample sizes it will be more dependent on kernel and hyperparameter choices adding a level of modeller input that may be unwanted — although calculation of the median heuristic removes a

level of subjectivity. Lastly, the area metric, although in the units of the quantity of interest, is relatively hard to objectively interpret. The area metric also displayed potential problems with not differentiating between under- and over-estimation of the variance for these numerical examples, often problematic when conservative results are required. It is noted that all the examples considered here have been for univariate distributions. Different conclusion may be found with higher-dimensional distributions in line with the findings of Aggarwal et al. where fractional norms increase sensitivity for high-dimensional non-statistical distances [66].

3.6 Conclusion

Validation is an important aspect of forward model-driven SHM which needs to be addressed before methods can be proposed. The objective of validation within forward model-driven SHM is to assess whether the predicted simulator distributions can be considered statistically similar to the observed distributions of damage states. In order to assess the inadequacy of simulators, given the target objective, a validation strategy has been proposed.

The proposed validation strategy implemented throughout this thesis begins with hypothesis testing. In a frequentist setting this will state whether there is statistical significance to reject the hypothesis that the simulator and observation damage state distributions are the same. In a Bayesian setting multiple hypotheses, set as multiple modelling approaches, can be compared in order to determine the ratio of posterior odds (or the Bayes factor). This approach means that the probability of a particular hypothesis can be determined and links into the problem model selection. The second stage of the strategy is to quantify the difference between two probability distributions using statistical distances. This provides a more informative measure of the simulator inadequacy. Thirdly, visual diagnostics such as the MMD witness function can be used to interpret the sources of these differences. Finally, if required, deterministic metrics such as residuals or Mean Squared Error (MSE) could be quantified.

Statistical distances have been compared using numerical examples. These case studies have led to the conclusion that the Kolmogorov distance is often insensitive to differences outside the central probability mass, making it impractical for some validation contexts. The KL-divergence will often be difficult to interpret, but can

provide useful information in diagnosing problems where significant differences (or impossibilities) in the probability mass are present. Both total variation and Hellinger distances show a good level of sensitivity to differences in distributions, with the total variation being more sensitive. In spite of this, it is the author's opinion that the Hellinger distance could be a more intuitive validation metric. The MMD distances produced similar distances to the Hellinger distance for these numerical example, meaning that it could be an informative and stable method for providing a non-parametric distance between samples. Finally, the area metric is useful in that it quantifies the distance in terms the quantity of interest units. Despite this the metric can be hard to objectively compare. Furthermore, it appears to fail to distinguish between under- and over-estimation of the variance. It is therefore suggested that for most validation applications a combination of the KL-divergence, area metric and either the total variation, Hellinger or MMD distances would be effective in assessing the simulator's adequacy.

Chapter 4

Emulators

Simulators are widely utilised throughout engineering to simulate behaviours of complex systems, with common methods being FEA, CFD and multi-physics techniques. The development of a simulator is often for a set of specific tasks, such as design space exploration, analysis or prediction. Increasingly simulators are being analysed using statistical methods e.g. [67–72] or used in optimisation routines e.g. [73–77], with both categories often requiring numerous simulator evaluations. This poses potential challenges as simulators are often computationally expensive, and for a set of inputs and parameters, outputs are unknown until the simulator is run. These challenges are common to forward model-driven SHM which employs simulators and statistical methods in order to produce health state outputs that are statistically representative of real world observations. Consequently for many optimisation or statistical methods to be practically applied to simulators emulators must be constructed. An emulator, surrogate model or meta-model, is a computationally efficient representation of the input-output mapping from a simulator, allowing fast evaluations in problems that require multiple runs of a simulator — with the drawback being that the simulator is approximated. To the author's knowledge the earliest example of constructing an emulator is Sacks et al. in 1989 [78, 79] where GP models were implemented.

The following chapter outlines the variety of approaches for constructing emulators with a discussion on why, in the author's opinion, for the majority of applications the most rigorous approaches are those based on GP technologies. A detailed overview of GP emulators is provided, examining the mathematics, implementation procedure, diagnostics and validation process. Additionally, techniques for efficiently emulating

large parameter spaces are demonstrated. Finally, additional extensions to GPs for a variety of emulation contexts are discussed.

4.1 Emulator Types

Simulators can be regarded as some functional mapping from a set of inputs $\boldsymbol{x}$ and parameters $\boldsymbol{\theta}$ to a set of outputs $\boldsymbol{y}$, $\eta : \{\boldsymbol{x}, \boldsymbol{\theta}\} \rightarrow \boldsymbol{y}$. This functional mapping, although based on mathematical equations, can often be considered mathematically intractable due to its complexity. This means that the function is treated as unknown, i.e. for a set of inputs and parameters, outputs are unknown until the simulator is run. In engineering problems deterministic simulators are most common, meaning that for a given set of inputs there will be a unique output, i.e. $y(\boldsymbol{x}) = \eta(\boldsymbol{x}, \boldsymbol{\theta})$. As a result, constructing a non-intrusive emulator $\hat{\eta}$, of a simulator is akin to that of regression, whereby an unknown functional mapping is approximated given a set of input-output data, $\mathcal{D}$. The data set $\mathcal{D}$ for building an emulator is a finite set of simulator runs. Thus, techniques for performing regression have been adopted for the purpose of emulation. There are several well established regression technologies that have been employed in constructing emulators, namely ANNs, Polynomial Chaos Expansion (PCE), Bayes Linear Analysis (BLA) and GPs. Section 4.1 proceeds with considering scenarios where parameters are removed from the problem, i.e. $y(\boldsymbol{x}) = \eta(\boldsymbol{x}) \approx \hat{\eta}(\boldsymbol{x})$, for simplicity of the discussion.

ANNs are a group of machine learning algorithms which aim to learn input-output mappings using combinations of computational units called neurons [26]. Each neuron takes a set of inputs, multiples these by a set of weights (and potentially adds bias terms) before applying a nonlinear activation function (popular choices being logistic or hyperbolic tangent functions) to generate the neuron's output. Neurons (or nodes) are structured given a topology. An ANN begins and ends with the inputs and outputs known as layers, for whom their size is determined by the inputs and outputs dimensionality. A topology specifies the number and size of hidden layers, intermediate stages between the input and output layers, including the connectivity of each node. The weights and bias terms for each node are grouped as the network parameters and are learnt in the training stage via minimisation of a loss function. This optimisation is usually performed using a form of gradient descent where the derivatives are calculated by back-propagation [26]. By specifying different topologies

an ANN should be able to capture any arbitrary function, given enough input-output data. On the other hand, an inappropriate topology can lead to overfitting [74], for example where the number of nodes in a hidden layer is very large. Additionally, to avoid overfitting model complexity must be regulated [73], which is often achieved by regularising the loss function or the addition of input noise [80].

ANNs have been employed as emulators in a variety of applications, examples being water management models [81, 82], aerodynamic CFD models [74], FEA models of crashworthiness [75] and structural damage [83] in addition to surrogate models of cost functions [76]. Frequently ANN emulators have been successfully utilised as part of optimisation tasks, generally within a deterministic framework [73–76, 81–83]. Several issues arise when using ANN emulators. Fitting topologies is a challenging but vital task in implementing ANNs, with a common approach being the trial and error method [75, 76, 82], however more advanced solutions have been developed, such as the NeuroEvolution of Augmenting Topologies (NEAT) method, that uses a Genetic Algorithm (GA) to learn optimal topologies of ANNs [84]. This process is time consuming, especially when employing the trial and error method which in addition requires a large amount of modeller input. Furthermore, ANNs produce a bias that is unpredictable [76], i.e. given a new set of input points it is unclear whether the emulator will over- or underestimate the functional response. ANNs are non-interpolating and consequently are not guaranteed to learn the underlying functional form. This is concerning in optimisation problems where ANN emulators can produce false maxima or minima [73, 81], with no indication of bias or whether the solution has overfit — leading to suboptimal parameter selection. Moreover, due to the problem of overfitting, an ANN requires cross-validation — involving dividing a data set into two sets, where one set is used to train and the other to validate the analysis. This process also increases the time cost in learning an ANN reflected in a paper by Broad et al. where training the ANN emulator alone took 16 hours compared to the 21 hours in performing the full optimisation task with the simulator [81].

Recently Deep Neural Networks (DNN)s have been implemented as emulators [72] for applications involving a large amount of training data where the *curse of dimensionality* — adding an extra dimension leads to an exponential increase in the functional input space, requiring a large increase in training data [26] — is problematic. Deep defines an ANN with numerous hidden layers, with development of DNNs improving problems with the curse of dimensionality by providing multiple layers of feature

extraction from the data. Nonetheless many of the issues associated with ANNs are still present in DNNs, such as overfitting and cross-validation of network topologies. Additionally, DNNs suffer from the problem of vanishing gradients, where the gradient of the loss function becomes vanishingly small, leading to weights that are updated by a negligible or zero term, which in turn leads to difficulties in training.

PCE is a methodology that can be applied to approximate the outputs of a simulator using finite series expansions. The approach can be implemented in an intrusive or non-intrusive manner for learning a set of unknown deterministic coefficients a_j. Access to the mathematical equations is required for intrusive PCE, as algebraic manipulations must be performed, meaning each implementation is complex and bespoke. This would require simultaneous development of both simulator and emulator. The aforementioned definition of a simulator, that it can be treated as intractable and therefore as a black box, in addition to the desire for a general emulation tool, means that intrusive PCE is excluded from this discussion of emulator types. Non-intrusive PCE therefore fits within the definitions as the approach uses evaluations of the simulator in order to infer a_j and can be directly compared with other emulator techniques.

Non-intrusive PCE assumes that the inputs are uncertain, represented as random variables X with joint PDF $f_X(\boldsymbol{x})$ resulting from marginally independent PDF's (a decorrelation step would be required if the output is also dependent on parameters [85]). Subsequently, assuming the uncertain outputs Y are a second-order stationary process, an expansion onto orthogonal polynomial bases is possible, as presented in Eq. (4.1).

$$Y \equiv \eta(X) \approx \sum_{j=0}^{p} a_j \boldsymbol{\psi}_j(X) \tag{4.1}$$

Where $\boldsymbol{\psi}_j(X)$ are specified multivariate polynomials that are completely dependant on the elicited joint PDF form of X (based on orthogonal properties), and p is the number of polynomials. If p were infinite then the expansion would be equivalent to Y, however practical implementation reduces this to a finite set of degrees not exceeding p. Examples of polynomial bases are the Legendre and Hermite polynomials, utilised when the input distribution form is uniform or Gaussian.

The coefficients a_j, can be determined non-intrusively using a variety of techniques, e.g.

a least squares regression or a quadrature approach (also known as the projection method) and once estimated the PCE emulator can be used to infer simulator outputs at new input locations. It has been shown that regression approaches lead to worse interpolation performance than quadrature methods [85]. In addition for situations with small sample sizes it has been recommended that twice over-determined regression should be used rather than uniquely-determined [86]. In comparison, quadrature methods involve solving complex integrals and as dimensions increase these become impractical due to the curse of dimensionality [87]. At this point computationally expensive sampling techniques may be used in order to estimate a_j. Challenges also remain, like in polynomial regression, in selecting the appropriate number of polynomials p. Normally the modeller will increase p until the new coefficients are small, and the approximation order large enough to provide adequate results. As PCE requires learning coefficients of orthogonal polynomials, overfitting becomes a problem, with cross validation methods required to increase generality outside the training data. Moreover, although error bounds on the approximation are available in some circumstances and estimates of the mean and variance of Y can be calculated, PCE provides no quantification of the uncertainty introduced by implementing the emulator [87], also known as code uncertainty. This means there is no clear mechanism in PCE formulations for expressing where the surrogate will perform poorly due to it being an approximation.

The use of PCE is most prevalent in the applied mathematics communities [87] with examples being soil foundation FEA [35], CFD [86, 88], probability of failure [89], fluid processes [36, 90], environmental systems [85], building performance [91] and structural mechanics [92] simulators. Predominantly PCE has been implemented in Uncertainty Propagation (UP) [85, 86, 88, 89, 92] and sensitivity analysis tasks [35, 36, 91]. Problems with generality, when PCE is applied to different input domains, are indicated in [91], displaying the issue of overfitting.

BLA provides a framework for performing analysis on problems too complex for standard Bayesian tools, pioneered by Goldstein and Wooff [93]. The approach updates simulator beliefs systematically using observational data via linear fitting; keeping the same form as Bayesian methods except dealing only with expectations, variances and covariances rather than probabilities. BLA supposes that there is some variable of interest B that we wish to infer given some other measured variable D. By specifying $\mathbb{E}(B)$, $\mathbb{E}(D)$, $\mathbb{V}(B)$, $\mathbb{V}(D)$, $\mathrm{cov}(B, D)$ and measuring D, it is possible to update the expectation and variance of B as shown in Eqs. (4.2) and (4.3). It is

noted that if all the prior quantities are specified as Gaussian then the BLA and full Bayesian solutions generate similar updating equations [94].

$$\mathbb{E}_D(B) = \mathbb{E}(B) + \mathrm{cov}(B, D)\mathbb{V}(D)^{-1}(D - \mathbb{E}(D)) \tag{4.2}$$

$$\mathbb{V}_D(B) = \mathbb{V}(B) - \mathrm{cov}(B, D)\mathbb{V}(D)^{-1}\mathrm{cov}(D, B) \tag{4.3}$$

It is typical in emulation using BLA to assume a prior functional form similar to that of Eq. (4.4).

$$y(X) = H\boldsymbol{\beta} + u(X) \tag{4.4}$$

Where H is a design matrix constructed from basis functions, $\boldsymbol{\beta}$ are regression coefficients and $u(X)$ is a discrepancy term from the basis fit. Priors in the forms of means and variances are then defined for the regression coefficients $\boldsymbol{\beta}$. The prior for the basis fit discrepancy term $u(X)$ is usually zero mean with a specified covariance structure, e.g. Eq. (4.5) [94]. The prior beliefs about the functional form can be formulated into a mean and covariance in Eqs. (4.6) and (4.7).

$$\mathrm{cov}(u(X), u(X')) = \sigma_u^2 \exp\left(-\theta_u (X - X')^{\mathsf{T}} (X - X')\right) \tag{4.5}$$

$$\mu(X) = \mathbb{E}(\eta(X)) = H\mathbb{E}(\boldsymbol{\beta}) \tag{4.6}$$

$$\kappa(X, X') = \mathrm{cov}(\eta(X), \eta(X')) = \mathrm{cov}(H(\boldsymbol{\beta}), H(\boldsymbol{\beta})) + \mathrm{cov}(u(X), u(X')) \tag{4.7}$$

The emulator mean and variance are then updated via Eqs. (4.2) and (4.3) using observed simulator outputs (i.e. $D = \boldsymbol{y}$). This generates the best linear fit for the emulator given a set of simulator runs, minimising the expected squared error loss. Furthermore, the variance in BLA provides a quantification of code uncertainty.

The key motives for applying BLA emulators are, that the distributional assumption

for the simulator outputs is unclear, and that a fully Bayesian analysis would produce computational complexities. Applications of BLA as emulators are fewer than the other emulator types with notable examples being in hydrocarbon reservoir pressures [67, 95], climate [70], galaxy formation [94, 96] and gas modelling [97, 98].

GP regression is a tractable Bayesian interpolation technique that is both a flexible and non-parametric method for inferring unknown functions from input-output data [27, 99]. A GP, a generalisation of the multivariate Gaussian distribution, can be considered a prior over functions, whereby inference is made in order to determine functions that were likely to have produced the output data. This means that a GP is the assumption that a finite set of variables are jointly Gaussian distributed. The Gaussian nature of the GP means that full Bayesian analysis remains tractable providing predictive equations for the expectation and variance in closed form (and therefore the full output PDF). All that is required to specify a GP are mean and covariance functions $m(\cdot)$ and $k(\cdot,\cdot)$ respectively; which state the modellers prior belief about the functional family that the simulator function may have been drawn from, presented in Eq. (4.8).

$$\eta(x) \sim \mathcal{GP}\left(m(X), k(X, X')\right) = \mathcal{N}\left(m(X), k(X, X')\right) \tag{4.8}$$

The Bayesian framework also provides protection against overfitting, having been demonstrated to have Occam's Razor (the selection of the minimally complex model) at work [18]. Full mathematical definitions are deferred to Section 4.2. It is noted that GPs and the term kriging are mathematically identical, with the kriging definition arising from low dimension geospatial applications [100].

The popularity of GP emulators in the statistics community reflects the rigorous statistical nature of the technique, with a tutorial by O'Hagan to help spread their uptake within engineering contexts [101]. Originally utilised by Sacks et al. [78, 79], GPs have been used in numerous applications with examples being chemical [68, 79], electronics [78, 102], spot weld [69], explosions in a cylinders [103], ecosystem [104], engineering design [77, 105, 106], climate [107, 108], disease [71, 109, 110] and dynamics [52, 55] simulators. In addition, GP models have been applied to outputs from non-smooth functions showing their applicability to a wide variety of simulators [111, 112].

It is clear from the aforementioned definitions and applications that both ANNs and

PCE have several shortcomings in their application as emulators. Firstly, ANNs and PCE have no mechanism for quantifying code uncertainty. As a consequence, the methods (in the forms commonly utilised in emulation) cannot inform the modeller of instances where the emulator is being utilised outside of the original training data. This may have severe implications as the modeller will be unaware of poor emulation, leading to suboptimal optimisation/calibration or unrealistic uncertainty propagation. In contrast BLA and GPs provide estimates of code uncertainty, which when incorporated into the statistical or optimisation method can provide resilience of the output to poor emulation, or even lead to opportunities for retraining and improving the emulator. It is argued that even when simulators are deterministic, probabilistic approaches will provide more robust outcomes. Secondly, ANNs and PCE can overfit if cross-validation strategies are not applied correctly, adding additional computational cost. On the other hand, GPs by construction have a built in mechanism for selecting the best, minimally complex model given the training observations, which in most contexts removes the need for cross validation. In addition, PCE and BLA are approximation methods, with PCE requiring a truncation level and BLA being a best linear estimate of a full Bayesian analysis. An ANN can also be considered an approximation, as an ANN with Gaussian priors on the weights will tend to a GP as the number of nodes in a hidden layer approaches infinity [113]. On the contrary, GPs are not an approximation and, as full Bayesian analysis, lead to an elegant closed form solution when the outputs are considered correlated and Gaussian. There may be instances where GPs may not be optimal, as when the output distribution cannot be assumed Gaussian, in these cases BLA may be preferred. However, in situations where the outputs are assumed Gaussian, GPs will be a more rigorous approach, with better representations of code uncertainty. In summary it is the author's opinion that for most applications GPs provide the most rigorous methodology for generating emulators and are the preferred emulator type in this thesis.

4.2 Gaussian Process Emulators

A simulator can be represented by an underlying functional mapping $\eta(\cdot)$ between a set of inputs X and their corresponding outputs Y, i.e. $Y = \eta(X)$. It may be possible to run the simulator at any arbitrary set of inputs, however to evaluate all the combinations of interest is assumed to be computationally expensive; for this

reason only a finite set of N simulator runs are often available. The objective of creating a GP emulator is to reproduce the functional mapping, via regression, so that predictions of the outputs can be made given new inputs. The probabilistic framework means that the mapping between a set of N inputs $X = \{\boldsymbol{x}_n\}_{n=1}^{N}$ of dimension D and their corresponding N outputs $\boldsymbol{y} = \{y_n\}_{n=1}^{N}$ is modelled as $p\left(\boldsymbol{y} \,|\, \boldsymbol{\eta}, X, \boldsymbol{\phi}\right)$; where $\boldsymbol{\phi}$ is a small set of hyperparameters (parameters of a prior distribution) and $\boldsymbol{\eta}$ are a vector of simulator output evaluations. For a GP emulator the latent function $\boldsymbol{\eta}$ can be modelled with a GP prior, this is based on the assumption that simulator outputs for different inputs can be modelled as jointly Gaussian distributed. The GP prior is formulated as presented in Eq. (4.9).

$$p\left(\boldsymbol{\eta} \,|\, X, \boldsymbol{\phi}\right) \sim \mathcal{N}\left(\boldsymbol{m}, K\right) \tag{4.9}$$

Where $\boldsymbol{m}$, the mean function, is linear in the parameters as demonstrated in Eq. (4.10) and K is the covariance function in Eq. (4.12).

$$\boldsymbol{m} = m(X) = H\boldsymbol{\beta} \tag{4.10}$$

The design matrix H is comprised of p basis functions, $H = (h_1(\cdot), \ldots, h_p(\cdot))$ applied to X, with p corresponding coefficients in the vector $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_p)^{\mathsf{T}}$. The basis functions used to construct H should reflect the prior beliefs about $\boldsymbol{\eta}$, as they form the assumption that $\boldsymbol{\eta}$ can be approximated by a function contained within the design matrix. It is common for a constant or linear set of basis functions to be used as this is often the degree of knowledge known about the simulator *a priori*. This formulation with explicit basis functions can also be constructed as in Eq. (4.11).

$$\eta(X) = m(X) + u(X) \tag{4.11}$$

This separates the prior into mean and covariance structures stating $u(X)$ as a zero-mean Gaussian process i.e. $\mathcal{GP}\left(\mathbf{0}\,, K\right)$. The covariance matrix K defines the prior assumption of the functions smoothness and is formed from the covariance function (also known as a kernel function), presented in Eq. (4.12). The covariance matrix, a description of the correlation between any two points in the input space via a RKHS, must be positive semi-definite.

Type	Function Name	Formulation
Mean	Constant	$\boldsymbol{m} = \mathbf{1}\beta$
Mean	Linear	$\boldsymbol{m} = \boldsymbol{x}\beta$
Mean	Polynomial	$\boldsymbol{m} = (\mathbf{1}, \boldsymbol{x}, \ldots, \boldsymbol{x}^p)^\mathsf{T}\boldsymbol{\beta}$ where p is the degree
Covariance	Squared Exponential (SE)	$K = \sigma_f^2 \exp\left(-(X - X')\Omega(X - X')^\mathsf{T}\right)$
Covariance	Matérn	$K = \sigma_f^2 \frac{2^{1-\nu}}{\Gamma(\nu)} \left(\frac{\sqrt{2\nu}\lvert X-X'\rvert}{\ell}\right)^\nu \mathcal{K}_\nu \left(\frac{\sqrt{2\nu}\lvert X-X'\rvert}{\ell}\right)$ where ν is a smoothness parameter commonly defines as a half integer i.e. $\nu = p + 1/2,\ p \in \mathbb{Z}$ and $\ell = 1/\omega$

Table 4.1: A selection of mean and covariance functions for GP emulators.

$$K = k(X, X') = \sigma_f^2 c(X, X'; \boldsymbol{\psi}) \tag{4.12}$$

Where σ_f^2 is an unknown variance, often called the scale factor or signal variance. A chosen correlation function $c(X, X'; \boldsymbol{\psi})$ will reflect the prior smoothness assumptions dependant on some hyperparameters $\boldsymbol{\psi}$. A natural choice for emulators is the Gaussian correlation function shown in Eq. (4.13), which assumes a smooth functional output typical for many simulators. This correlation function leads to a Squared Exponential (SE) kernel, which is also a stationary covariance invariant to translations in the inputs.

$$c(X, X') = A = \exp\left(-(X - X')\Omega(X - X')^\mathsf{T}\right) \tag{4.13}$$

Where $\Omega = \mathrm{diag}(\omega_1, \ldots, \omega_D)$ is a diagonal matrix of roughness parameters, defining an Automatic Relevance Determination (ARD) correlation function [27]. ARD kernels scale the effect each input dimension, where large values indicate a long term trend for that dimension. Consequently, a covariance function depends on a set of hyperparameters $\boldsymbol{\phi}_k = \{\sigma_f^2, \boldsymbol{\psi}\}$ which for a SE ARD kernel are $\{\sigma_f^2, \omega_1, \ldots, \omega_D\}$. A selection of mean and covariance functions are presented in Table 4.1. The selection of a particular mean and covariance function must reflect prior assumptions about the structure of the simulator output [114].

The joint prior between the latent function values (for training $\boldsymbol{\eta}$ and testing $\boldsymbol{\eta}_*$) at training and testing inputs, X and X_* respectively, can be formed as in Eqn. 4.14; this uses the definition that a GP is collection of random variables where a finite set

has a joint Gaussian distribution[1].

$$\begin{bmatrix} \boldsymbol{\eta} \\ \boldsymbol{\eta}_* \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \boldsymbol{m}_\eta \\ \boldsymbol{m}_* \end{bmatrix}, \begin{bmatrix} K_{\eta,\eta} & K_{\eta,*} \\ K_{*,\eta} & K_{*,*} \end{bmatrix} \right) \tag{4.14}$$

The likelihood for a GP emulator is typically modelled as Gaussian, however the variance term of the likelihood is debated. It is common for a GP emulator to assume that the observations are 'noise-free' [78], i.e. repeats at the same set of inputs will always result in the same output - for a deterministic simulator. Even so, due to numerical instabilities in inverting the covariance matrix, GP regression becomes impractical unless a nugget term is added; usually a fixed small number to the diagonal of the covariance matrix i.e. $\tilde{K}_{\eta,\eta} = (K_{\eta,\eta} + \nu\mathbb{I})$; this is discussed further in Section 4.2.1. The following mathematical definitions include a nugget term ν, resulting in $p(\boldsymbol{y} \,|\, \boldsymbol{\eta}) = \mathcal{N}(\boldsymbol{\eta}, \nu\mathbb{I})$ as the likelihood.

In order to perform inference the joint prior is combined with the likelihood to form the joint posterior $p(\boldsymbol{\eta}, \boldsymbol{\eta}_* \,|\, \boldsymbol{y}, \boldsymbol{\phi})$ (the inputs are dropped for simplicity of notation) using Bayes' theorem (see Appendix A.1). The latent training function, $\boldsymbol{\eta}$ can then be marginalised out, using standard multivariate Gaussian conditioning, to form the posterior in closed form as shown in Eqs. (4.15) to (4.17) (see Appendix A.2 for derivations).

$$p(\boldsymbol{\eta}_* \,|\, \boldsymbol{y}, \boldsymbol{\phi}) = \mathcal{N}(\mathbb{E}_1(\boldsymbol{\eta}_*), \mathbb{V}_1(\boldsymbol{\eta}_*)) \tag{4.15}$$

$$\mathbb{E}_1(\boldsymbol{\eta}_*) = H_* \boldsymbol{\beta} + K_{*,\eta} \tilde{K}_{\eta,\eta}^{-1} (\boldsymbol{y} - H_\eta \boldsymbol{\beta}) \tag{4.16}$$

$$\mathbb{V}_1(\boldsymbol{\eta}_*) = K_{*,*} - K_{*,\eta} \tilde{K}_{\eta,\eta}^{-1} K_{\eta,*} \tag{4.17}$$

Equations (4.15) to (4.17) describe the full predictive equations and are often used in the machine learning literature [27], however this formulation leaves $\{\boldsymbol{\beta}, \sigma_f^2, \boldsymbol{\psi}\}$ as the set of hyperparameters to be inferred. Following the work of Bastos and O'Hagan

[1]For compactness $\boldsymbol{m}_a$ is the mean function relating to the latent function a and $K_{a,b}$ is the covariance matrix between the latent functions a and b, e.g. $K_{\eta,*}$ is the covariance between the training and testing latent functions.

[115] a weak (standard non-informative) prior is specified for $\boldsymbol{\beta}$ and σ_f^2 known as the Jeffreys prior, $p\left(\boldsymbol{\beta}, \sigma_f^2\right) \propto 1/\sigma_f^2$. This information, combined with Eq. (4.9) using Bayes' theorem, leads to an analytical posterior for $(\boldsymbol{\beta}, \sigma_f^2)$, a normal inverse-gamma distribution, constructed from Eqs. (4.18) and (4.19).

$$p\left(\boldsymbol{\beta} \,|\, \boldsymbol{y}, \sigma_f^2, \boldsymbol{\psi}\right) \sim \mathcal{N}\left(\hat{\boldsymbol{\beta}}, \sigma_f^2\, W\right) \tag{4.18}$$

Where $\hat{\boldsymbol{\beta}} = W_\eta\, H_\eta^\mathsf{T}\, \tilde{A}_{\eta,\eta}^{-1}\, \boldsymbol{y}$ and $W = (H_\eta^\mathsf{T}\, \tilde{A}_{\eta,\eta}^{-1}\, H_\eta)^{-1}$.

$$p\left(\sigma_f^2 \,|\, \boldsymbol{y}, \boldsymbol{\psi}\right) \sim \mathcal{IG}\left(\frac{N-p}{2}, \frac{(N-p-2)\,\hat{\sigma}_f^2}{2}\right) \tag{4.19}$$

Where $\hat{\sigma}_f^2 = (\boldsymbol{y} - H_\eta\, \hat{\boldsymbol{\beta}})^\mathsf{T}\, \tilde{A}_{\eta,\eta}^{-1}\, (\boldsymbol{y} - H_\eta\, \hat{\boldsymbol{\beta}}) / (N-p-2)$. The hyperparameters $\boldsymbol{\beta}$ can be marginalised out by integrating the product of Eq. (4.15) and Eq. (4.18) with respect to $\boldsymbol{\beta}$ resulting in Eqs. (4.20) to (4.22).

$$p\left(\boldsymbol{\eta}_* \,|\, \boldsymbol{y}, \sigma_f^2, \boldsymbol{\psi}\right) = \mathcal{N}\left(\mathbb{E}_2\left(\boldsymbol{\eta}_*\right), \mathbb{V}_2\left(\boldsymbol{\eta}_*\right)\right) \tag{4.20}$$

$$\mathbb{E}_2\left(\boldsymbol{\eta}_*\right) = H_*\, \hat{\boldsymbol{\beta}} + A_{*,\eta}\, \tilde{A}_{\eta,\eta}^{-1}\, (\boldsymbol{y} - H_\eta\, \hat{\boldsymbol{\beta}}) \tag{4.21}$$

$$\mathbb{V}_2\left(\boldsymbol{\eta}_*\right) = \sigma_f^2 \left(A_{*,*} - A_{*,\eta}\, \tilde{A}_{\eta,\eta}^{-1}\, A_{\eta,*} + PWP^\mathsf{T}\right) \tag{4.22}$$

Where $P = H_* - A_{*,\eta}\, \tilde{A}_{\eta,\eta}^{-1}\, H_\eta$. Following the same procedure as $\boldsymbol{\beta}$, σ_f^2 is integrated out from the product of Eq. (4.20) and Eq. (4.19), leading to a Student's t process, as presented in Eqs. (4.23) to (4.25).

$$p\left(\boldsymbol{\eta}_* \,|\, \boldsymbol{y}, \boldsymbol{\psi}\right) = \mathcal{TP}\left(N-p, \mathbb{E}\left(\boldsymbol{\eta}_*\right), \mathbb{V}\left(\boldsymbol{\eta}_*\right)\right) \tag{4.23}$$

$$\mathbb{E}\left(\boldsymbol{\eta}_*\right) = H_*\, \hat{\boldsymbol{\beta}} + A_{*,\eta}\, \tilde{A}_{\eta,\eta}^{-1}\, (\boldsymbol{y} - H_\eta\, \hat{\boldsymbol{\beta}}) \tag{4.24}$$

$$\mathbb{V}\left(\boldsymbol{\eta}_*\right) = \hat{\sigma}_f^2 \left(A_{*,*} - A_{*,\eta}\, \tilde{A}_{\eta,\eta}^{-1}\, A_{\eta,*} + PWP^{\mathsf{T}} \right) \tag{4.25}$$

Full Bayesian analysis would require setting a prior and integrating out $\boldsymbol{\psi}$ from the posterior, in order to obtain the uncertainty associated with $\boldsymbol{\psi}$. Due to the correlation matrices A, and $\hat{\sigma}_f^2$ depending on $\boldsymbol{\psi}$, integrating out $\boldsymbol{\psi}$ leads to a highly intractable function. This could be calculated numerically using a sampling approach, typically via an MCMC algorithm at a high computational cost. In contrast, a Maximum Likelihood Estimate (MLE), generated by maximising the marginal likelihood (usually framed for numerical reasons as minimising the Negative Log Marginal Likelihood (NLML), i.e. Eq. (4.26) is optimised according to Eq. (4.27)), is a fast and efficient form of inference with the only negative being that the posterior variance is slightly underestimated [69, 116]. Typically the computational savings outweigh the negative, especially when emulating smooth simulators [69]. For this reason the plug-in approach is implemented in this thesis.

$$-\log p\left(\boldsymbol{\psi} \mid \boldsymbol{y}\right) \propto \frac{1}{2}\log\left|A_{\eta,\eta}\right| - \frac{1}{2}\log\left|W\right| + \frac{N-p}{2}\log(\hat{\sigma}_f^2) \tag{4.26}$$

$$\hat{\boldsymbol{\psi}} = \arg\min(-\log p\left(\boldsymbol{\psi} \mid \boldsymbol{y}\right)) \tag{4.27}$$

Equation (4.27) in practice is implemented as an optimisation problem, typically using a gradient based approach [27]. Rogers et al. demonstrate that a global optimisation, specifically a quantum particle swarm differential evolution technique, consistently optimises to better space when compared to a conjugate gradient approach [117]. As a result in this thesis optimisation of the marginal likelihood is performed using a quantum particle swarm algorithm [118].

A visual representation of the GP prior and Bayesian updated posterior (conditioned on two observations) are presented in Fig. 4.1. The example shows a GP with a zero mean and SE covariance function prior, which as stated is a prior over functions. Samples can be drawn from the prior and Fig. 4.1a presents 20 draws indicating possible plausible functions specified by the prior. Next, the simulator is evaluated at two input points, providing two observations of the unknown latent function. These evaluations are used in Eq. (4.23) performing a Bayesian update on the prior. This can also be thought of as conditioning a joint multivariate Gaussian

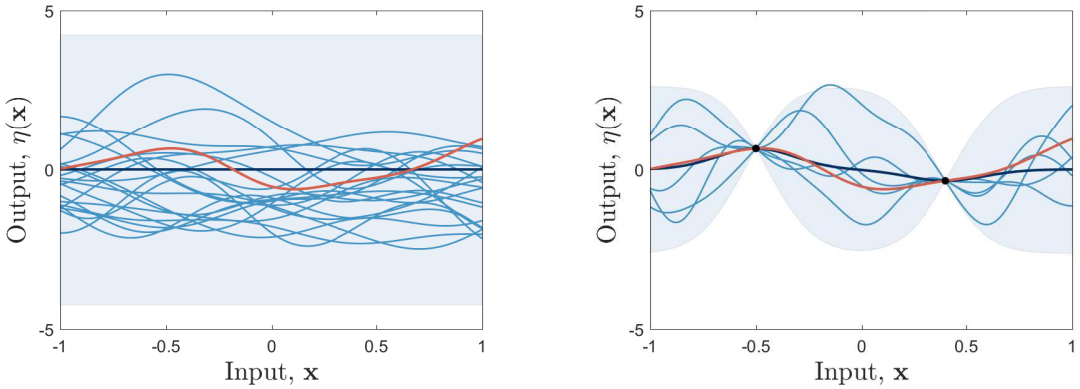


Figure 4.1: Bayesian update of a univariate GP. Panel (a) demonstrates a zero mean GP prior, with mean (dark blue) and three standard deviation confidence intervals (blue shaded region), alongside samples from the prior (blue) and the simulator (red). Panel (b) presented the posterior update given two simulator outputs as observations $(\cdot)$, with mean (dark blue), three standard deviation confidence intervals (blue shaded region), posterior samples (blue) and the simulator (red)

on two observations. Figure 4.1b shows the posterior distribution of the functions given two outputs, which fit the simulator function exactly at those points with no code uncertainty; expected given a deterministic simulator. Away from the observations, code uncertainty (indicated by the 3σ confidence intervals) increases in a smooth manner consistent with the covariance function selection (here a SE is used) conditioned on the MLE estimates of the hyperparameters $\boldsymbol{\phi}_k$. The posterior mean has altered fitting the data points in a smooth manner and tending towards the prior mean as predictions occur away from the observations. This means that as a GP emulator predicts away from a trained input region, predictions will return to the prior. As more simulator observations are added, the GP will improve and code uncertainty will reduce.

4.2.1 Numerical Issues

A key assumption in generating a probabilistic GP emulator for a deterministic simulator is that the GP will fit a known simulator observation exactly with no code uncertainty - a 'noise-free' assumption. This follows naturally as given the same inputs the same output will always occur for a deterministic simulator. As a consequence, no noise model is included in a GP emulator.

Within other fields, such as machine learning and spatial modelling (kriging), where real world observations are used, it is common to apply a homoscedastic noise model, represented by a likelihood $p(\boldsymbol{y} \mid \boldsymbol{\eta}) = \mathcal{N}(\boldsymbol{\eta}, \sigma_n^2 \mathbb{I})$ [27, 100]. This amounts to including a constant on the diagonal of the covariance matrix, inferred as an extra parameter in optimising the NLML. The inclusion of a noise model has the effect of smoothing through observation points whilst estimating a noise variance. The predicted posterior variance is a combination of the noise and latent function variance (code uncertainty in an emulator context). An additional by-product of including a noise variance is that it provides numerical stability in inverting the covariance (or correlation matrices) in Eqs. (4.24) to (4.26).

It is debated in the field of emulation whether a nugget term ν is appropriate. The 'noise-free' assumption clearly argues that the term should not be included. Nonetheless in practical implementation, correlation (or covariance) matrices can become ill-conditioned and, as the roughness parameters tend to infinity, become so poorly conditioned that inversion is not possible. A frequently utilised pragmatic solution is the addition of a very small nugget term to alleviate these problems. However, adding a nugget is known to have the same affect as the noise variance, sometimes leading to inference of the latent function $\boldsymbol{\eta}$ that smooths through observations with a small variance, meaning that the GP no longer performs exact interpolation. Per contra, it can be argued that the inclusion of a nugget captures any discrepancy between the GP and the simulator, which may arise due to inaccurate assumptions about stationarity, covariance structure or because certain inputs have been excluded from the training set but will have a small effect on the simulator output.

Andrianakis and Challenor proposed a penalty to the marginal likelihood in order to force a GP emulator with a nugget term to fit known data points exactly [116]. This technique removes the NLML mode associated with type II likelihoods - those that approximate the function, smoothing through and treating the nugget as noise - and keeps the type I mode - the interpolation solution - when a nugget term is applied. The penalty term, presented in Eq. (4.28), is a ratio of the MSE between emulator mean and training outputs $\bar{M}(\boldsymbol{\psi}, \nu)$, and the MSE between a least squares estimate and training outputs $\bar{M}(0)$.

$$\pi(\omega, \nu) = \exp\left(-2\frac{\bar{M}(\boldsymbol{\psi}, \nu)}{\varepsilon \bar{M}(0)}\right) \tag{4.28}$$

Where $\bar{M}(\boldsymbol{\psi}, \nu) = (\nu^2/N)(\boldsymbol{y} - H_\eta\,\hat{\boldsymbol{\beta}})^\mathsf{T}\, A_{\eta,\eta}^{-2}\,(\boldsymbol{y} - H_\eta\,\hat{\boldsymbol{\beta}})$ and $\bar{M}(0) = (\boldsymbol{y} - H_\eta\,\hat{\boldsymbol{\beta}}_0)^\mathsf{T}(\boldsymbol{y} - H_\eta\,\hat{\boldsymbol{\beta}}_0)/N$, where $\hat{\boldsymbol{\beta}}_0 = (H_\eta^\mathsf{T}\,H_\eta)^{-1}H_\eta^\mathsf{T}\,\boldsymbol{y}$ - the least squares estimate. The interval $[0, -2]$ is chosen by Andrianakis and Challenor due to its association with 95% of the probability mass of a distribution [116]. The parameter ε affects the penalisation amount, with a larger portion of the marginal likelihood excluded as $\varepsilon \to 0$. An optimal heuristic is $\varepsilon = 1 \times 10^{-3}$, which excludes the type II mode whilst keeping the type I mode [116]. In this thesis a nugget term is included to improve the conditioning of $A_{\eta,\eta}$ whilst penalising the NLML as shown in Eq. (4.29).

$$-\log p\,(\boldsymbol{\psi} \mid \boldsymbol{y}) \propto \frac{1}{2}\log|A_{\eta,\eta}| - \frac{1}{2}\log|W| + \frac{N-p}{2}\log(\hat{\sigma}_f^2) + 2\frac{\bar{M}(\Omega,\nu)}{\varepsilon\bar{M}(0)} \tag{4.29}$$

Furthermore, the main computational load of training and predicting with a GP is the inversion of $A_{\eta,\eta}$ which has time complexity $\mathcal{O}(N^3)$ often performed via a Cholesky decomposition (Appendix A.3). By storing the inverted correlation matrix, $A_{\eta,\eta}^{-1}$ prediction then becomes $\mathcal{O}(N)$ and $\mathcal{O}(N^2)$ for the mean and covariance respectively. Resultantly, as the data set size N increases the computational load may become very burdensome and memory issues may also occur. Solutions to these problems are addressed in Section 4.3.

4.2.2 Validation and Diagnostics

Discussed in Chapter 3, validation is an important process in the construction of any model type, with GP emulators being no exception. A GP can provide poor emulation of a simulator for two main reasons. Firstly, the model form assumed by the initial GP prior is not appropriate for the simulator's functional form. This can occur if any component of the prior is ill-suited to the functional structure of the simulator. For example, the mean function could be inappropriate, e.g. if a polynomial mean is used for a periodic simulator output. The covariance function could also impose incorrect assumptions, for example a stationary kernel is employed when the correlation is input dependent (where some regions have a faster functional transition that others) or if a SE kernel is utilised when the simulator output is non-smooth. These problems are often solved with better model selection. On the other hand, if joint normality is an unreasonable assumption for the simulator outputs, and no transform of the output distribution possible, or if the outputs are

uncorrelated, then a GP may be an invalid assumption. A second reason is when the training data used to estimate the hyperparameters does not fully reflect, or is a restricted case of the general space in which the emulator is expected to generalise. This may lead to poor estimates of the hyperparameters that do not generalise. In addition, if MLEs of the hyperparameters are employed (as stated in Section 4.2) rather than integrating them out of the full posterior, and these estimates are in the tails of the hyperparameter distribution due to inappropriate training data, the posterior prediction will be poor as it is conditioned on these hyperparameters. As a result diagnostics are required in order to improve predictions and generate a valid emulator for a specific task.

In the following section diagnostic tools, validation metrics and their specific application to GP emulators are presented. These tools are implemented on a numerical example which is displayed in Fig. 4.2. The simulator equation is shown in Eq. (4.30), where a grid of ten ($N = 10$) evenly spaced evaluations are used as training data $\mathcal{D} = \{\boldsymbol{x}, \boldsymbol{y}\}$ for the emulator. Validation data is constructed from the inputs $\boldsymbol{x}_* = \{-1, 0.99, \ldots, 1\}$ and their corresponding outputs $\boldsymbol{y}_*$. The GP prior is formed from a linear mean function $H = 1$ of dimension $p = 1$ and a SE kernel. The MLE estimates of the hyperparameters are: $\hat{\omega} = 50.83$, $\hat{\sigma}_f^2 = 0.84$ and $\beta = 1.27$ (with a fixed nugget $\nu = 1 \times 10^{-8}$).

$$\boldsymbol{y} = \eta(\boldsymbol{x}) = 2\cos(2\pi \times 2.5\boldsymbol{x}) - 2.5\cos(2\pi \times 2.2\boldsymbol{x}) + 0.15\cos(2\pi \times 6\boldsymbol{x}) + 5\boldsymbol{x} - 2 \quad (4.30)$$

Individual Prediction Error (IPE) (or standardised residuals) allow an input dependant assessment of the emulator predictions and are formulated via Eq. (4.31).

$$D_{IPE}(\boldsymbol{y}_*) = \frac{\boldsymbol{y}_* - \mathbb{E}\left(\boldsymbol{\eta}_*\right)}{\sqrt{\text{diag}(\mathbb{V}\left(\boldsymbol{\eta}_*\right))}} \quad (4.31)$$

When the posterior is constructed from Eq. (4.23), and the emulator represents the simulator well, these residuals should be distributed as a standard Student t distribution of $N - p$ degree of freedoms (conditioned on the training data $\mathcal{D}$ and hyperparameters $\boldsymbol{\psi}$). As the number of data points and degrees of freedom increase, (or the posterior formulation in Eq. (4.15) is implemented) then the standardised residuals will tend to a Gaussian distribution. The degrees of freedom $(N - p)$ equal

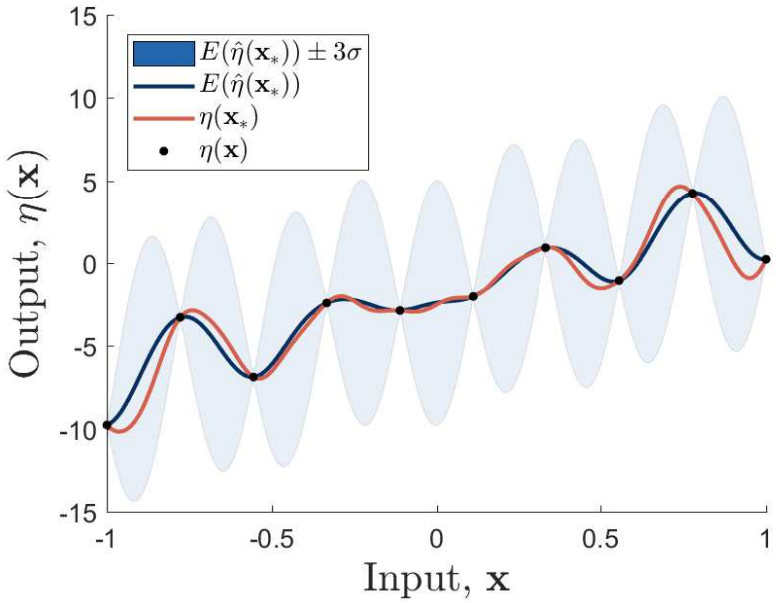
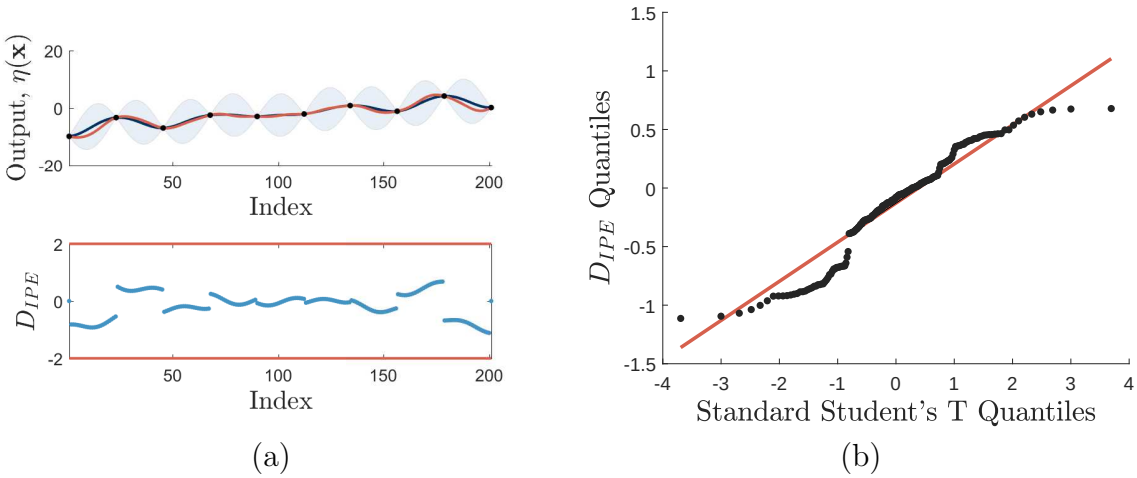


Figure 4.2: Posterior emulator prediction for the diagnostics numerical example.

9 from the training data in Fig. 4.2 meaning that the standardised residuals should tend to a Student's t. Visually a QQ-Plot, where the quantiles of two distributions are compared, can indicate whether the residuals are Student's t distributed. Fig. 4.3 presents two graphical interpretations of IPE, a comparison with the input index (Fig. 4.3a), and a QQ-Plot (Fig. 4.3b).

IPE should remain low, with large values indicating problems, and clusters of high standardised residuals indicating a systematic failure. To diagnose the cause several avenues must be explored. When values are large and clustered close to validation points the problem may be that the roughness parameters Ω are too small and have been poorly inferred due to an inadequate training data set. If the values are large but no systematic patterns can be determined, the estimate of the signal variance σ_f^2 is likely to be the problem. When a large number of high IPEs are shown and they are of the same sign, the mean function and $\boldsymbol{\beta}$ coefficient have been inappropriately specified or a non-stationary kernel should be used. A heuristic definition of 'large' can be $|D_{IPE}| > 2$ [115] (the same definition is used for all residual diagnostics). Figure 4.3a therefore indicates that the residuals are appropriate. It can be seen that the locations of worst performance are near the ends of the function, with a clear patterns visible. This indicates that although the emulator is adequate in terms of IPE residuals, the functional form has not been fully captured, potentially due to the roughness parameter in the SE kernel. More training points around the ends of



(a) (b)

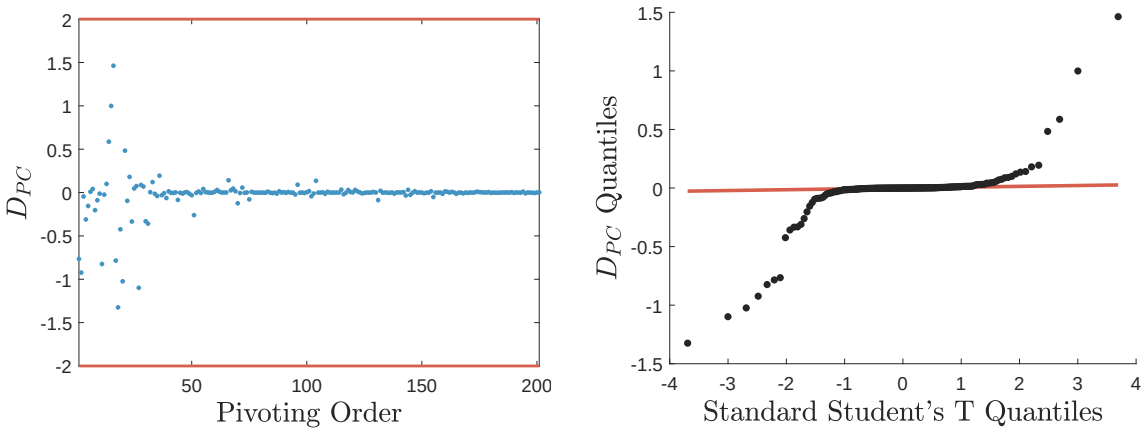
Figure 4.3: IPE diagnostic. Panel (a) displays the diagnostic against the input index with ±2 thresholds and panel (b) a QQ-Plot of the residuals.

the function, or a change in covariance function to a Matérn kernel (that captures less smooth functions better, see Table 4.1 — when $\nu = \infty$, Matérn = SE) could be possible remedies. The QQ-Plot indicates that the IPEs are approximately standard Student's t distributed, however the area of steeper gradient indicates a possible underestimation of the signal variance σ_f^2.

Another method for generating standardised residuals, that removes correlation unlike IPEs, are variance decomposition approaches. Here a standard deviation matrix G, capturing the cross terms, is generated from a decomposition of the posterior covariance matrix, i.e $\mathbb{V}(\boldsymbol{\eta}_*) = GG^\mathsf{T}$. The standardised errors, with uncorrelated elements and unit variances are formed from Eq. (4.32).

$$D_{VD}(\boldsymbol{y}_*) = G^{-1}(\boldsymbol{y}_* - \mathbb{E}(\boldsymbol{\eta}_*)) \tag{4.32}$$

These residuals, in the same manner as IPEs, indicate if the normality assumption is invalid, being standard Student's t distributed with $N - p$ degrees of freedom when Eq. (4.23) is used. This means that a QQ-Plot can be used as a graphical test. Likewise as with IPEs, large values and systematic patterns diagnose problems with the emulator, however their interpretation will change based on the decomposition. Various decompositions can be used such as an eigenvalue, Cholesky or a pivoted Cholesky decomposition. Figure 4.4 presents a demonstration of the diagnostic using a pivoted Cholesky decomposition. By sorting the validation data by largest variance, conditioned on the previous element, the pivoted approach produces the permutation



(a) (b)

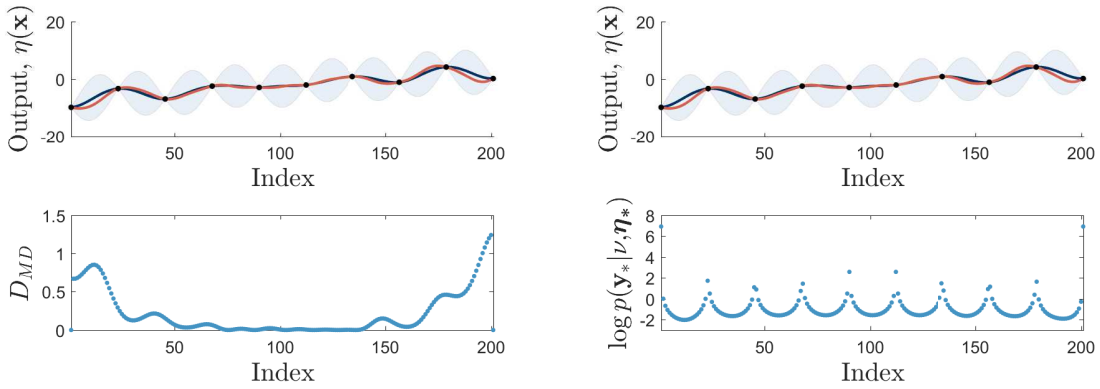
Figure 4.4: Standard residuals via a pivoted Cholesky decomposition. Panel (a) displays the diagnostic against the pivoting order from the decomposition with ±2 thresholds and panel (b) a QQ-Plot of the residuals.

matrix P and the upper triangular matrix R where the standard deviation becomes $G = PR^{\mathsf{T}}$. If the standardised residuals are high/low in the initial part, this indicates that σ_f^2 is inappropriate or that the function is heterogeneous, whereas the end of the standardised residuals will indicate issues with the covariance function structure and/or roughness parameters Ω. Figure 4.4a, although presenting residuals within the thresholds, evidences that the variance of the function could be better captured. This is stated by the large D_{PC} at the beginning of the pivot order and may infer that either σ_f^2 is underestimated or that the roughness parameter ω is too long. This is confirmed by the QQ-Plot where although the majority of data lies on the reference line, heavy tails indicate larger variability than estimate by the emulator. When compared with IPEs, it can be seen that the cause of this seems to be because the estimated emulator function smooths through the first and last simulator points.

Mahalanobis distances can be implemented as a summary statistic or diagnostic and is a measure of the distance between a point and an ellipse. The metric can be formulated from the variance decomposition, $D_{MD}(\boldsymbol{y}_*) = D_{VD}(\boldsymbol{y}_*)^{\mathsf{T}} D_{VD}(\boldsymbol{y}_*)$ or from Eq. (4.33).

$$D_{MD}(\boldsymbol{y}_*) = (\boldsymbol{y}_* - \mathbb{E}(\boldsymbol{\eta}_*))^{\mathsf{T}} \mathbb{V}(\boldsymbol{\eta}_*)^{-1} (\boldsymbol{y}_* - \mathbb{E}(\boldsymbol{\eta}_*)) \tag{4.33}$$

If the emulator outputs are fully independent then the Mahalanobis distance with the full posterior covariance $\mathbb{V}(\boldsymbol{\eta}_*)$ and variance $\text{diag}(\mathbb{V}(\boldsymbol{\eta}_*))$ matrices will be equal.



(a) (b)

Figure 4.5: Panel (a) presents the individual Mahalanobis distances D_{MD}, when independence of the outputs is assumed. Panel (b) displays the individual log posterior likelihoods $p\left(\boldsymbol{y}_* \mid \nu, \boldsymbol{\eta}_*\right)$ (where $\nu = N - p$, the degrees of freedom), when independence of the outputs is assumed.

A comparison of these two values can help diagnose the degree of correlation in the posterior; 11.47 and 42.28 respectively for the example indicate a degree of dependence in the outputs. A large Mahalanobis distance would indicate poor emulation of the simulator.

Individual Mahalanobis distances can be calculated by assuming the posterior at each test point is independent. A visualisation of this metric presents a scale of the distance between posterior predictions and test points. Figure 4.5a demonstrates the application of this diagnostic. It can clearly be seen that predictions are worst at the edges of the function, again stating that the variation of the simulator has not been fully captured, indicating the roughness parameter may be too long.

The posterior density is a scaled version of the Mahalanobis distance. Here the posterior PDF is assessed for the predictive point; interpreted as a posterior likelihood — the likelihood of the test point being drawn from the model. The PDF will either be Gaussian or Student's t depending on the posterior equations. When Eq. (4.23) is implemented the diagnostic is formulated as shown in Eqs. (4.34) and (4.35).

$$p\left(\boldsymbol{y}_* \mid N - p, \boldsymbol{\eta}_*\right) = Z_t \left(1 + \frac{1}{N-p}(\boldsymbol{y}_* - \mathbb{E}\left(\boldsymbol{\eta}_*\right))^{\mathsf{T}} \mathbb{V}\left(\boldsymbol{\eta}_*\right)^{-1} (\boldsymbol{y}_* - \mathbb{E}\left(\boldsymbol{\eta}_*\right))\right)^{-\frac{N-p+n}{2}} \tag{4.34}$$

$$Z_t = \frac{\Gamma\left(\frac{N-p+n}{2}\right)}{\Gamma\left(\frac{N-p}{2}\right)}(N-p)^{-n}(\pi)^{-n/2}|\mathbb{V}\left(\boldsymbol{\eta}_*\right)|^{-1/2} \tag{4.35}$$

Where n is the number of test points in $\boldsymbol{y}_*$, Z_t is the normalising constant and $\Gamma(\cdot)$ is a gamma function. For numerical reasons the diagnostic is often used in log form, called the log posterior likelihood (i.e. $\log p\left(\boldsymbol{y}_* \mid N-p, \boldsymbol{\eta}_*\right)$). Both dependent and independent forms can be calculated (in the same fashion to the Mahalanobis distance), however the log posterior likelihood is generally less interpretable than the Mahalanobis distance. For the numerical example the log posterior likelihood is 1366 and -117 when $\mathbb{V}\left(\boldsymbol{\eta}_*\right)$ and $\text{diag}(\mathbb{V}\left(\boldsymbol{\eta}_*\right))$ are used respectively. This would further indicate that the test data is most likely to have come from the emulator with dependence between the outputs, meaning the independence assumption is not maintained. Furthermore, the log posterior likelihood can be calculated individually, assuming independent outputs (i.e. using $\text{diag}(\mathbb{V}\left(\boldsymbol{\eta}_*\right))$), as shown in Fig. 4.5b. The log posterior likelihood is high at the training points and decreases away from these locations, with global minimums around the start and end of the function. This reinstates that the emulator performs poorly in these regions.

Model criticism via MMD is also achievable (see Section 3.4 for mathematical details) [49], however it is excluded from this discussion as it is best suited for situations where the model is compared to stochastic outputs.

Standard regression diagnostics that treat the output as deterministic can also be implemented. These measures will fail to fully explain the performance of a GP emulator due to its probabilistic formulation and therefore should never solely be used to assess the emulator validity. These scores are primarily useful for comparing GPs with other deterministic regression approaches and assessing the posterior mean. Two diagnostics are presented: NMSE and R^2 score.

The NMSE formulation, presented in Eq. (4.36), is a highly interpretable diagnostic. A score of zero indicates mean predictions without any error. Conversely, a score of 100 represents a scenario where the prediction is no better than taking the mean of the true values ($\bar{\boldsymbol{y}}_* = \mathbb{E}\left(\boldsymbol{y}_*\right)$).

$$NMSE = \frac{100}{N\sigma^2_{\boldsymbol{y}_*}} \sum (\boldsymbol{\eta}_* - \boldsymbol{y}_*)^2 \tag{4.36}$$

Where $\sigma^2_{\boldsymbol{y}_*}$ is the variance of test outputs. The mean prediction from the emulator has a NMSE of 6.43, which implies a relatively good fit.

The R^2 score (or coefficient of determination) is a measure of how well the mean prediction explains the test points variation (a ratio of the model explained variation over the total variation). The R^2 score can be calculated using Eq. (4.37), where a score of zero indicates that 0% of the variation is explained by the model and 1 represents that 100% of the variation is captured.

$$R^2 = 1 - \frac{\sum(\boldsymbol{y}_* - \boldsymbol{\eta}_*)^2}{\sum(\boldsymbol{y}_* - \bar{\boldsymbol{y}}_*)^2} \tag{4.37}$$

An issue arises that as more basis functions are added to H, the R^2 score will always increase, meaning that the score will improve with overfitting. Instead the adjusted R^2 should be used to take into account the degrees of freedom of the data and the model, shown in Eq. (4.38).

$$aR^2 = 1 - (1 - R^2)\frac{N-1}{N-p-1} \tag{4.38}$$

This addition means that the score is penalised as more basis functions are added, meaning that it will favour minimally complex models. The adjusted R^2 score for the example is 0.93, and would indicate that the emulator mean captures the variation in test outputs well.

By considering all the diagnostics presented, the emulator in Fig. 4.2 can be shown to be functionally appropriate, with the simulator test data lying comfortably within the predicted probability mass. Improvements could be made in order to better capture the beginning and end of the function. These improvements could be to change the covariance function to a Matérn class, to increase the training data with evaluations near the beginning and end of the function or to improve the estimates of σ_f^2 and ψ.

4.2.3 Latin Hypercube Design

GP emulators are constructed from a set of N simulator evaluations. However, due to the computational expense in running a simulator each evaluation should be optimal

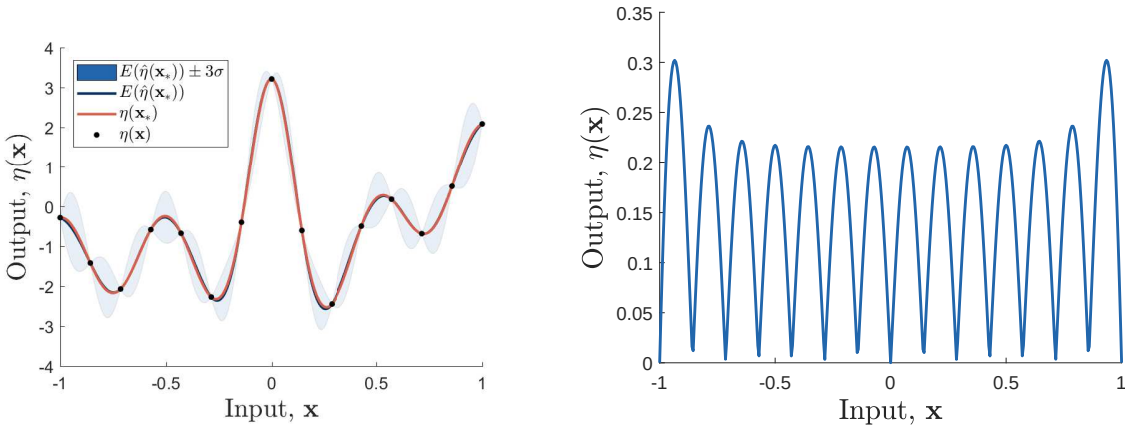
for making inferences about the simulator. The process of generating a strategy for where to evaluate a simulator sits within the Design of Experiments (DoE) field. The main objective of a DoE is to fill a given input domain, known as space-filling. In the context of a simulator it may be that several parameters are to be statistically studied and require emulation. This leads to designing an experiment that covers a several dimension sized domain in which simulator evaluations are to be run. A DoE method will look to fill that space in a manner that allows good coverage for a given budget of simulator runs.

For the majority of emulation applications an initial space filling design is required (that may later be updated in order to improve emulator performance). Numerous strategies exist for generating a DoE with examples being Monte Carlo sampling techniques, Latin Hypercube Design (LHD), maximum entropy sampling (discussed further in Section 6.2.2), Sobol sampling and Halton sampling. Detailed explanations of these approaches are beyond the scope of this thesis, with the choice of DoE method being user and problem dependent; for a detailed review see [119]. Most of these approaches create a uniformly spaced design, however when fitting a GP emulator, evaluation locations should also be close to the domain boundary in order to accurately capture the behaviour in these regions. To visualise this problem an example is introduced where a simulator is constructed from Eq. (4.39) (with 15 equidistant training points) and is presented in Fig. 4.6a (the hyperparameter estimates are $\hat{\omega} = 30.15$, $\hat{\sigma}_f^2 = 1.42$ and $\hat{\beta} = 0.66$ with a fixed nugget $\nu = 1 \times 10^{-9}$). Typical code uncertainty will be in the form of Fig. 4.6b, where increases are seen around the boundary of the domain, meaning that to improve emulator performance a concentration of design points should be located at the boundary. A method for achieving this is called a Generalised Maximum Latin Hypercube Design (GMLHD) [120].

$$\boldsymbol{y} = \eta(\boldsymbol{x}) = 1.2\boldsymbol{x} + \mathcal{N}\left(\boldsymbol{x} \,|\, 0, 0.1\right) - \mathcal{N}\left(\boldsymbol{x} \,|\, 0.3, 0.3\right) - \mathcal{N}\left(\boldsymbol{x} \,|\, -0.1, 0.4\right) + \cos(2\pi \times 2\boldsymbol{x}) \tag{4.39}$$

Latin Hypercubes

A Latin Hypercube (LHC) is a random space filling DoE that is a D dimensional extension of the Latin square sampling method. A sampling design is Latin square if



(a) (b)

Figure 4.6: Panel (a) presented the posterior emulator prediction for a numerical example. Panel (b) demonstrates typical GP emulator uncertainty (one standard deviation, $\sqrt{\text{diag}(\mathbb{V}(\boldsymbol{\eta}_*))}$) when equidistant training points are used in $[-1, 1]$.

given an $N \times N$ grid of possible sample locations in two dimensions ($D = 2$), there is only one sample in each row and location. An example for $N = 21$ is displayed in Fig. 4.7a.

To construct a random LHC L, in the space $\mathbb{R}^D$ of N points (in each dimension), elements of the vector $\boldsymbol{x} = \{x_1, \ldots, x_N\}^\mathsf{T}$ (typically in $[0, 1]$ and then scaled) are transformed through random permutations for each dimension (i.e. $L = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_D\}$). However, by construction a LHC will not necessarily be maximally separated. For this reason maximum (or optimised) LHCs are constructed.

Optimal criteria must be defined in order to generate a maximum LHC. Here two criteria are used, a distance measure (Eq. (4.40)) — specifically the LHC with minimum squared euclidean distance $d(L)$ and minimal re-occurrences of that minimum distance $n(L)$ — and a force measure (Eq. (4.41)) — namely the sum norm of the repulsive forces $F(L)$, when samples are considered electrically charge particles (where a squared term is used to avoid square root computations, increasing the speed with which $F(L)$ is calculated).

$$d(L) = \min_{1 \leq i,\, j \geq N,\, i \neq j} ||\boldsymbol{x}_i - \boldsymbol{x}_j||^2 \tag{4.40}$$

$$F(L) = \sum_{i=1}^{N} \sum_{j=i+1}^{N} \frac{1}{||\boldsymbol{x}_i - \boldsymbol{x}_j||^2} \tag{4.41}$$

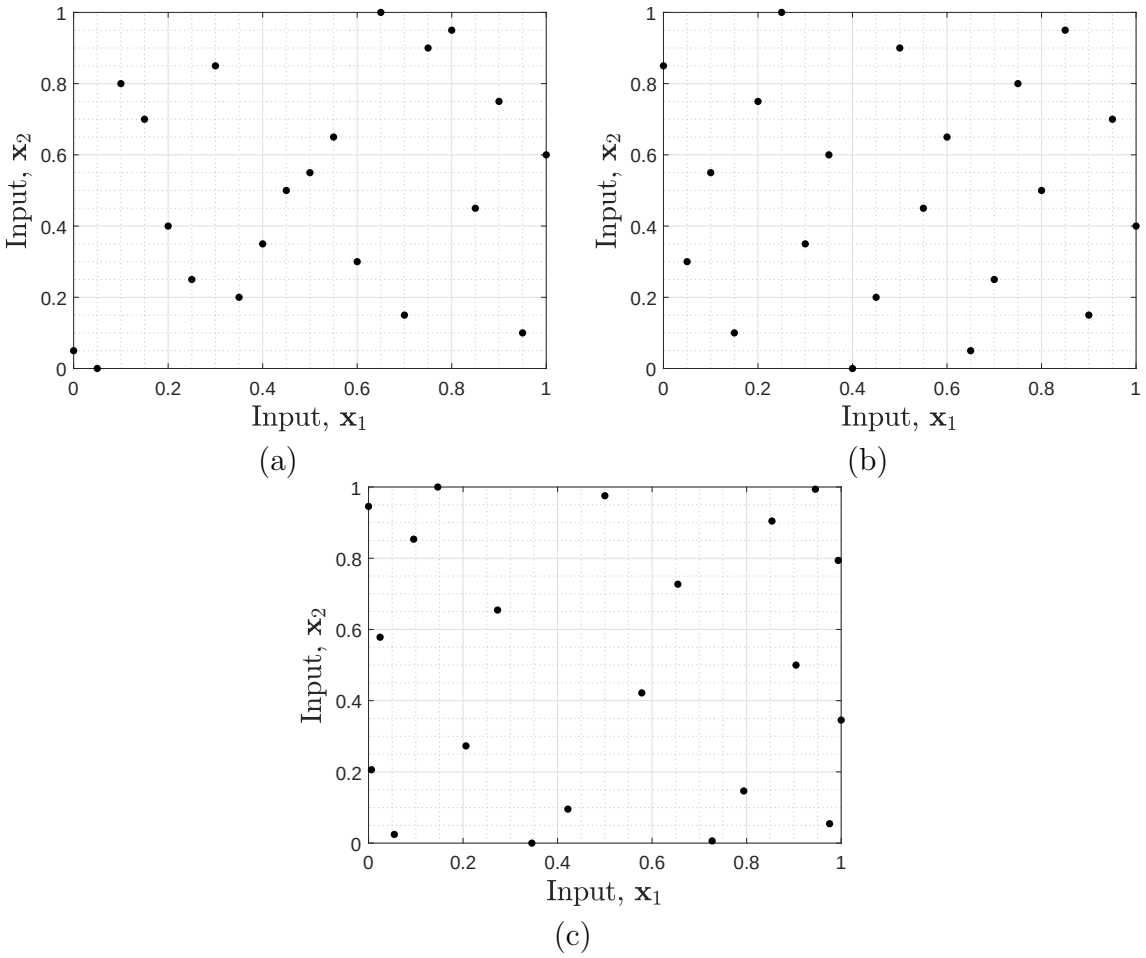


Figure 4.7: Latin squares where $N = 21$. Panel (a) demonstrates a random Latin square, panel (b) a maximum Latin square (using a force criteria), and panel (c) a generalised maximum Latin square (transforming (b) through Eq. (4.43) where $a = 0$).

With these definitions a LHC L_1 is better than L_2 when $d(L_1) > d(L_2)$; in scenarios where $d(L_1) = d(L_2)$ then where $n(L_1) < n(L_2)$. For the force criteria, L_1 is better than L_2 when $F(L_1) < F(L_2)$. The specified criteria can be framed as an optimisation problem to identify a maximum LHC for $\boldsymbol{x}$ in $\mathbb{R}^D$.

One approach to optimising a LHC is to use a genetic algorithm [121] as outlined in Algorithm 2. The fitness function is either evaluated by assessing $d(L)$ and $n(L)$ (if distance is the criteria) or $F(L)$ (when force is used). The best half of the population are the largest distances (with the least repeats of that distance) or the smallest force, with these surviving LHCs becoming parents. In the cross-over stage, children are created by keeping the best LHC (becoming the 1st and $(N_{pop}/2+1)$th child) and performing cross-overs with the remaining i survivors. The first $N_{pop}/2$ children are

obtained by taking a random column of the ith parent and substituting it into the best LHC. Remaining children are generated by taking a random column of the best LHC and placing it inside the corresponding ith parent. Once all the children have been generated mutation is performed on all but the 1st child. Here each column for every child is assigned a number in $[0, 1]$ based on a uniform distribution and when lower than a threshold p_{mut}, two random elements are swapped in that column before the fitness is assessed. The best LHC from that new population is checked against Eq. (4.42) (for the force criteria). An example of an optimised LHC is presented in Fig. 4.7b.

$$F(L_k) - F(L_{k-n}) < \varepsilon(F(L_n) - F(L_0)) \tag{4.42}$$

Where n is a few iterations (e.g. $n = 50$), as there is no guarantee of improvement every iteration, L_0 is the initial best LHC, k is the current iteration where Eq. (4.42) is only assessed when k is a multiple of n, and ε is small (here $\varepsilon = 10^{-7}$).

Algorithm 2 Optimised Maximum Latin Hypercube

```
Draw N_pop random LHCs
Evaluate fitness for all individuals in population
Stop = false
while Stop ≠ true do
    Select best half of population as survivors
    Cross-over survivors to generate N_pop children
    Mutate children to generate new population
    Evaluate fitness for all individuals in new population
    if best LHC meets stopping criteria then
        Stop = true
    end if
end while
```

Generalised Maximum Latin Hypercube Design

A GMLHD aims to reduce the uncertainty at the edges of a GP emulator by placing design points near the boundary, whilst remaining well spaced [120]. The main approach is to take a uniform maximum LHC (in $[0, 1]^D$) where the ith, jth element is denoted $z_{i,j}$ and transform the design points through a beta quantile function (an inverse CDF) given a tuning parameter $a \in [0, 1]$, shown in Eq. (4.43).

LHD	NMSE	D_{MD}	$\log p\left(\boldsymbol{y}_* \mid N-p, \boldsymbol{\eta}_*\right)$
Random	90.281 ± 62.162	5078 ± 1.7739	-186 ± 420
Maximum	0.012	338	1494
Generalised ($a = 0.8$)	0.008	345	1507
Generalised ($a = 0.6$)	0.005	292	1502
Generalised ($a = 0.4$)	0.004	206	1474
Generalised ($a = 0.2$)	0.003	142	1429
Generalised ($a = 0$)	0.003	113	1374

Table 4.2: Comparison of GP emulator predictions when trained using different LHDs where $D = 2$ and $N = 21$. The random LHD results are an average of 25 realisations with the mean and standard deviation are shown. The simulator is Eq. (4.44). Both D_{MD} and $\log p\left(\boldsymbol{y}_* \mid N-p, \boldsymbol{\eta}_*\right)$ assume independent posterior variance.

$$x_{i,j} = \frac{1}{\mathcal{B}((1+a)/2, (1+a)/2)} \int_0^{z_{i,j}} t^{(a-1)/2}(1-t)^{(a-1)/2} dt \tag{4.43}$$

Where $\mathcal{B}$ denotes a beta function. A beta quantile function is implemented as it is known that for large degree polynomial regression an arc-sine distribution (when $a = 0$) is the limit distribution of its D-optimal design (see [120] for more mathematical justifications). An arc-sine distribution will put more mass on the design space edges, whereas the other extreme where $a = 1$ will result in a uniform distribution (leaving the maximum LHC unchanged). Figure 4.7c presents an example of a generalised maximum Latin square where Figure 4.7b is transformed through Eq. (4.43).

Table 4.2 presents a comparison of LHDs when a random, maximum and generalised maximum LHCs are used to determine the training points of a GP emulator, where $D = 2$ and $N = 21$. The numerical example uses the simulator shown in Eq. (4.44). The training GP emulators were tested against a $N \times N$ grid and validation metrics assessed as displayed in Table 4.2.

$$\boldsymbol{y} = \eta(X) = 2(\boldsymbol{x}_1 - 2 + 10\boldsymbol{x}_2 - 8\boldsymbol{x}_2^2)^2 + 2\sqrt{\boldsymbol{x}_2 + 1}\,(2\boldsymbol{x}_2)^2 \tag{4.44}$$

It is demonstrated that as expected a random LHD performs worst on all validation metrics with a maximum LHD being outperformed by the GMLHD. This agrees with finding of Dette and Pepelyshev in [120], who show that a GMLHD will outperform a maximum LHD and Sobol sampling for a variety N and D. Generally the decrease in a coincides with better emulator performance, as shown by the NMSEs and

Mahalanobis distances. On the contrary, the log posterior likelihood indicates that the GP model which most likely explains the data is that trained using a GMLHD where $a = 0.8$. The parameter a should be set according to the intended use case, with a low a being applied in scenarios where the function needs to be accurately specified at the boundary with low variance.

4.3 Sparse Gaussian Process Emulators

An issue with GP regression is that training costs $\mathcal{O}(N^3)$ (for N observations) with prediction of the mean and variance costing both $\mathcal{O}(N)$ and $\mathcal{O}(N^2)$ respectively [122–124]. Although substantially more computationally efficient than running a simulator, this time complexity can make GPs computationally demanding in circumstances where N is significantly large, such as in large parameter spaces, often due to high-dimensionality. Sparse GP approximations seek to reduce this computational complexity by reducing the computational load of inverting $K_{\eta,\eta}$.

The simplest and most naive approach is to select a Subset of Data (SoD) of size Q from the full training data set (of size N where $Q \ll N$) in order to scale down the time complexity to $\mathcal{O}(Q^3)$ [122]. The problem is difficult as it relies on a known redundancy within the original data set, which is often not the case — especially in expensive evaluations of a simulator. This loss of information is generally unacceptable in an emulation context, as any simulator runs are expected to have come at a large computational cost. An alternative to SoD is the local GP approach [123]. A simple implementation of local GPs is to divide a data set into equal block sizes of size B and fit a GP to each block; reducing the computational complexity to $\mathcal{O}(NB^2)$. An issue with the technique is that discontinuities will occur between each data block, which can be unacceptable in emulators that are assumed to have smooth outputs. A less naive implementation of the local GP approach is to use a clustering algorithm to categorise the data into various subsets and fit GP models to each subset of data. As a consequence, the computational complexity of the method will not only be dominated by the largest subset, but will also incur the additional cost of the clustering algorithm. As a result, both SoD and local GP approaches are often not appropriate.

Two key approaches exist for generating sparse GPs, approximating the model or posterior. The techniques use inducing inputs [122] (originally referred to as ‘pseudo-

inputs' [125]) $Z = \{\boldsymbol{z}_m\}_{n=1}^{M}$, that have latent function outputs $\boldsymbol{u}$ (realisations of a GP), known as inducing variables, in order to produce sparsity. These two groups of methods are discussed in the following sections. It is noted that the approaches below are conditional on the full set of hyperparameters $\boldsymbol{\phi}$, where similar procedures to those in Section 4.2 could be used to marginalise them out. A zero mean function is also assumed in order to simplify notation. Furthermore, both the X and $\boldsymbol{\phi}$ are dropped from the conditional probabilities in order to preserve neatness and interpretability of notation.

4.3.1 Model Approximations

Quiñonero-Candela and Rasmussen present a unified framework for model approximations [122]. These approaches seek to modify the joint prior $p(\boldsymbol{\eta}_*, \boldsymbol{\eta})$ of the GP Eq. (4.14) in order to replace the complexity of inverting $K_{\eta,\eta}$ with a less expensive inversion. This is performed by incorporating inducing points $\{Z, \boldsymbol{u}\}$ (where Z are a set of inducing inputs and $\boldsymbol{u}$ are the corresponding latent function evaluations) into the joint prior $p(\boldsymbol{\eta}_*, \boldsymbol{\eta}, \boldsymbol{u})$ and marginalising the inducing variables, $\boldsymbol{u}$, out of the posterior (although Z will affect the final solution). The key assumption for these sparse methods is that the joint prior can be approximated by assuming conditional independence between $\boldsymbol{\eta}_*$ and $\boldsymbol{\eta}$ given $\boldsymbol{u}$. This means that $\boldsymbol{\eta}_*$ and $\boldsymbol{\eta}$ are only linked through $\boldsymbol{u}$; demonstrated in Eq. (4.45).

$$p(\boldsymbol{\eta}_*, \boldsymbol{\eta}) \simeq q(\boldsymbol{\eta}_*, \boldsymbol{\eta}) = \int p(\boldsymbol{\eta}_* \,|\, \boldsymbol{u}) q(\boldsymbol{\eta} \,|\, \boldsymbol{u}) p(\boldsymbol{u}) d\boldsymbol{u} \tag{4.45}$$

Where $p(\boldsymbol{u}) = \mathcal{N}(0, K_{u,u})$ is the prior[2] for the latent variables $\boldsymbol{u}$ and the test conditional, $p(\boldsymbol{\eta}_* \,|\, \boldsymbol{u})$, is defined in Eq. (4.46).

$$p(\boldsymbol{\eta}_* \,|\, \boldsymbol{u}) = \mathcal{N}\left(K_{*,u}K_{u,u}^{-1}\boldsymbol{u}, K_{*,*} - Q_{*,*}\right) \tag{4.46}$$

It is noted that the notation $Q_{a,b} = K_{a,u}K_{u,u}^{-1}K_{u,b}$ is used. The two model approximation methods detailed differ in their assumption about the training conditional $q(\boldsymbol{\eta} \,|\, \boldsymbol{u})$, whilst assuming the same prior for the inducing variables and likelihood.

[2]It is common for a nugget, $\varepsilon\mathbb{I}$ to be incorporated here [126] for the same reasons as outline for emulators previously, i.e. increases the stability of the inversion of the covariance matrix. A nugget is implemented in this thesis meaning $p(\boldsymbol{u}) = \mathcal{N}(0, K_{u,u} + \varepsilon\mathbb{I})$.

The assumptions for the training conditional $q(\boldsymbol{\eta} \mid \boldsymbol{u})$, and the marginalised joint prior $p(\boldsymbol{\eta}_*, \boldsymbol{\eta})$, for both a Deterministic Training Conditional (DTC) and Fully Independent Training Conditional (FITC) approximation are shown in Table 4.3. The main difference between DTC and FITC is clear in the joint prior, presented in Table 4.3. The top left corner of the covariance is modified in FITC so that the approximation includes the exact covariance on the diagonal. This transforms the training conditional from deterministic to fully independent.

Method	**DTC**	**FITC**
$q(\boldsymbol{\eta} \mid \boldsymbol{u})$	$\mathcal{N}\left(K_{\eta,u}K_{u,u}^{-1}\boldsymbol{u}, 0\right)$	$\mathcal{N}\left(K_{\eta,u}K_{u,u}^{-1}\boldsymbol{u}, \mathrm{diag}(K_{\eta,\eta} - Q_{\eta,\eta})\right)$
$p(\boldsymbol{\eta}_*, \boldsymbol{\eta})$	$\mathcal{N}\left(\mathbf{0}, \begin{bmatrix} Q_{\eta,\eta} & Q_{\eta,*} \\ Q_{*,\eta} & K_{*,*} \end{bmatrix}\right)$	$\mathcal{N}\left(\mathbf{0}, \begin{bmatrix} Q_{\eta,\eta} - \mathrm{diag}(Q_{\eta,\eta} - K_{\eta,\eta}) & Q_{\eta,*} \\ Q_{*,\eta} & K_{*,*} \end{bmatrix}\right)$

Table 4.3: DTC and FITC assumptions for the training conditional $q(\boldsymbol{\eta} \mid \boldsymbol{u})$ and the joint prior $p(\boldsymbol{\eta}_*, \boldsymbol{\eta})$. The joint prior $p(\boldsymbol{\eta}, \boldsymbol{\eta}_*)$ is calculated by substituting the training condition $q(\boldsymbol{\eta} \mid \boldsymbol{u})$ into Eq. (4.45) and solving the integral which can be done in closed form.

The posteriors $q(\boldsymbol{\eta}_* \mid \boldsymbol{y})$ and log marginal likelihoods $p(\boldsymbol{y} \mid X)$ for the DTC and FITC approximations can be unified into the analytical form outlined in Eq. (4.47) and Eq. (4.48) [124]. This is performed by substituting the assumptions from Table 4.3 into Eq. (4.45) and solving the integral (using standard Gaussian conditionals in Appendix A.2).

$$q(\boldsymbol{\eta}_* \mid \boldsymbol{y}) = \mathcal{N}\left(Q_{*,\eta}\bar{K}_{\eta,\eta}^{-1}\boldsymbol{y}, K_{*,*} - Q_{*,\eta}\bar{K}_{\eta,\eta}^{-1}Q_{\eta,*}\right) \tag{4.47}$$

$$\log p(\boldsymbol{y} \mid X) = -\frac{1}{2}\log|\bar{K}_{\eta,\eta}| - \frac{1}{2}\boldsymbol{y}^T\bar{K}_{\eta,\eta}^{-1}\boldsymbol{y} - \frac{N}{2}\log 2\pi \tag{4.48}$$

Where $\bar{K}_{\eta,\eta} = Q_{\eta,\eta} + \mathrm{diag}(\alpha(K_{\eta,\eta} - Q_{\eta,\eta})) + \nu\mathbb{I}$. The marginal likelihood and posterior of the two methods can be formulated by setting α to zero or one for the DTC and FITC approximations respectively. After setting α, the low rank structure of $\bar{K}_{\eta,\eta}$ should be exploited using the Woodbury inversion and determinant lemmas in order to improve the computational efficiency (see Appendix A.3). These amendments reduce the computational complexity for training to $\mathcal{O}(NM^2)$ and for prediction to $\mathcal{O}(M)$ and $\mathcal{O}(M^2)$ for the mean and variance respectively [122–124].

The inducing inputs can be either a subset of the input data or as any set of points from the real line. The subset of inputs poses challenges when global prediction quality is required as the selection of inducing inputs from a discrete set of data will involve some form of greedy or combinatorial optimisation. In contrast, considering the inducing inputs as any point on the real line leads to a continuous optimisation problem [125]. This allows the inducing inputs to be inferred via optimisation of the log marginal likelihood. When the inducing inputs are equal to the training inputs, the marginal likelihood and the posterior are the same as the full GP for both DTC and FITC. A key drawback of model approximation methods are that optimising via the approximate marginal likelihood means treating the inducing inputs as parameters of the model, adding all the problems of overfitting and optimisation that are evident in parametric models [124, 127]. This view of the inducing points means the assumptions about the data and inference approximations are coupled. Learning via the exact marginal likelihood of the approximate model also means that the hyperparameters will be optimal for the approximate model and not necessarily the full GP.

Figures 4.8 and 4.9 present univariate numerical examples where the simulator output is a sample from a GP process with zero mean and a SE covariance; $\sigma_f^2 = 1$ and $\omega = 8$. The examples demonstrate the difference in the two approaches when the hyperparameters $\boldsymbol{\phi}$ and inducing inputs Z are learnt through optimising the log marginal likelihood in Eq. (4.48). These illustrate a comparison of the two sparse GP methods, DTC and FITC, with a full GP solution and the training data, where the mean and $\pm 3\sigma$ confidence intervals are displayed for the full and sparse GPs. It is shown that FITC gives a better approximation of the variance than DTC that tends to overestimate (due to the deterministic assumption). Signs of overfitting are present in both methods. The variance for DTC when $X \approx 0.9$ reduces almost to zero, displaying overconfidence in the prediction when it would be expected to increase from the last training point, shown in the full GP solution. Figure 4.9 visually demonstrates that FITC fits the middle section of training data well, however the variance starts to increase before the training data boundary. This indicates that the inducing points have been placed in locations that overfit the middle section of the training data, leading to poor generalisation.

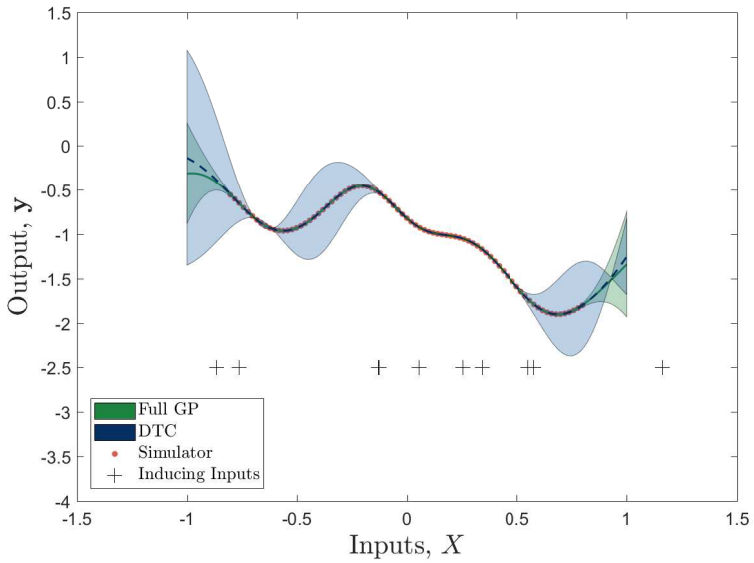


Figure 4.8: Predictions from a sparse DTC GP with 10 inducing points, against a full GP and training simulator data for a numerical example. Shaded regions indicate $\pm 3\sigma$ confidence levels.

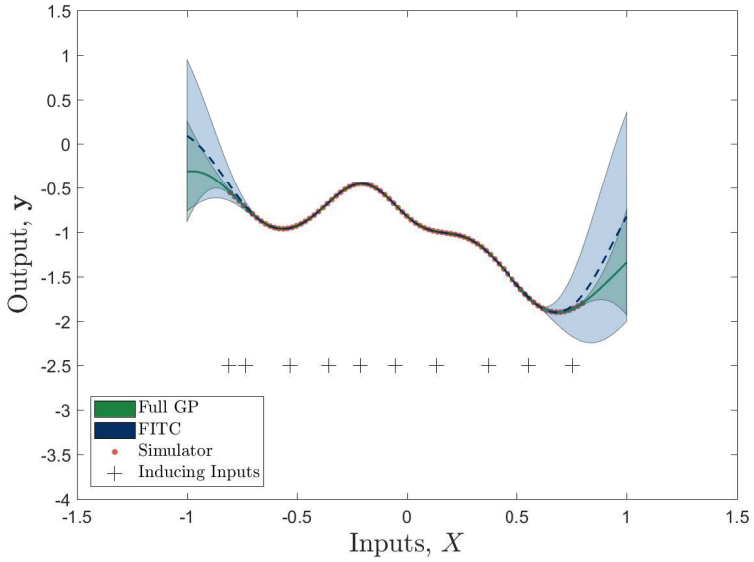


Figure 4.9: Predictions from a sparse FITC GP with 10 inducing points, against a full GP and training simulator data for a numerical example. Shaded regions indicate $\pm 3\sigma$ confidence levels.

4.3.2 Posterior Approximations

An alternative approach to model approximations is to apply sparsity at the inference stage, approximating the posterior and marginal likelihood. Here two approaches Variational Free Energy (VFE) [127] and Power Expectation Propagation (PEP) are considered — PEP has been shown to be a framework unifying both VFE and FITC [124].

VFE aims to approximate the true posterior directly by constructing a variational approximation and maximising the evidence lower bound $F_v(Z, \phi)$ (which is a lower bound of the log marginal likelihood $\log p(\boldsymbol{y} \mid X)$) using Jensen's inequality. VFE is a specific form of variational inference (also formulated in a more general sense with an uncollapsed bound [128]) that incorporates the inducing inputs as parameters of the variational inference removing problems associated with treating them as model parameters.

The approach begins by describing the predictive posterior in Eq. (4.15) as the marginalisation of the conditional prior $p(\boldsymbol{\eta}_* \mid \boldsymbol{\eta})$, $p(\boldsymbol{\eta}_* \mid \boldsymbol{y}) = \int p(\boldsymbol{\eta}_* \mid \boldsymbol{\eta}) p(\boldsymbol{\eta} \mid \boldsymbol{y}) \mathrm{d}\boldsymbol{\eta}$ which becomes the target of a variational approximation. By augmenting the integral with a set of inducing variables $\boldsymbol{u}$, with the assumption that $\boldsymbol{\eta}_*$ and $\boldsymbol{\eta}$ are conditionally independent given $\boldsymbol{u}$, (in a similar fashion to a FITC model approximation approach) an approximate predictive posterior is formed as in Eq. (4.49); where $\phi(\boldsymbol{u}) = \mathcal{N}(\boldsymbol{\mu}_v, \Sigma_v)$, a 'free' variational Gaussian distribution dependant on the 'free' inputs Z.

$$p(\boldsymbol{\eta}_* \mid \boldsymbol{y}) \approx q(\boldsymbol{\eta}_*) = \int p(\boldsymbol{\eta}_*, \boldsymbol{u}) \phi(\boldsymbol{u}) \mathrm{d}\boldsymbol{u} = \int q(\boldsymbol{\eta}, \boldsymbol{u}) \mathrm{d}\boldsymbol{u} \tag{4.49}$$

The inducing inputs Z and the 'free' distribution $\phi(\boldsymbol{u})$ can be specified by minimising the divergence between the variational distribution and the augmented true posterior distribution $p(\boldsymbol{\eta}, \boldsymbol{u} \mid \boldsymbol{y})$, using the KL-divergence, $KL(q(\boldsymbol{\eta}, \boldsymbol{u}) || p(\boldsymbol{\eta}, \boldsymbol{u} \mid \boldsymbol{y}))$ — this is equivalent to maximising the lower bound of the true log marginal likelihood defined in Eq. (4.50) and rearranged in Eq. (4.51).

$$F_v(Z, \phi) = \int p(\boldsymbol{\eta} \mid \boldsymbol{u}) \phi(\boldsymbol{u}) \log \frac{p(\boldsymbol{y} \mid \boldsymbol{\eta}) \cancel{p(\boldsymbol{\eta} \mid \boldsymbol{u})} p(\boldsymbol{u})}{\cancel{p(\boldsymbol{\eta} \mid \boldsymbol{u})} \phi(\boldsymbol{u})} \mathrm{d}\boldsymbol{\eta} \mathrm{d}\boldsymbol{u} \tag{4.50}$$

$$F_v(Z,\phi) = \int \phi(\boldsymbol{u}) \left(\int p\left(\boldsymbol{\eta} \,|\, \boldsymbol{u}\right) \log p\left(\boldsymbol{y} \,|\, \boldsymbol{\eta}\right) \mathrm{d}\boldsymbol{\eta} + \log \frac{p\left(\boldsymbol{u}\right)}{\phi(\boldsymbol{u})} \right) \mathrm{d}\boldsymbol{u} \tag{4.51}$$

Where Eq. (4.51) separates out the integral with respect to $\boldsymbol{\eta}$, which can be solved in Eq. (4.52) and substituted into Eq. (4.51) forming Eq. (4.53).

$$\int p\left(\boldsymbol{\eta} \,|\, \boldsymbol{u}\right) \log p\left(\boldsymbol{y} \,|\, \boldsymbol{\eta}\right) \mathrm{d}\boldsymbol{\eta} = \log \left(\mathcal{N}\left(K_{\eta,u} K_{u,u}^{-1} \boldsymbol{u}, \mathbb{I}\nu \right) \right) - \frac{1}{2\nu} \mathrm{tr}(K_{\eta,\eta} - Q_{\eta,\eta}) \tag{4.52}$$

$$F_v(Z,\phi) = \int \phi(\boldsymbol{u}) \log \frac{\mathcal{N}\left(K_{\eta,u} K_{u,u}^{-1} \boldsymbol{u}, \mathbb{I}\nu \right) p\left(\boldsymbol{u}\right)}{\phi(\boldsymbol{u})} \mathrm{d}\boldsymbol{u} - \frac{1}{2\nu} \mathrm{tr}(K_{\eta,\eta} - Q_{\eta,\eta}) \tag{4.53}$$

The logarithm can be moved outside of the integral in Eq. (4.53) by assuming that the Jensen's inequality (the assumption that formed Eq. (4.50)) can be reversed, leading to the $\phi(\boldsymbol{u})$ terms cancelling. The variational lower bound is therefore formed in Eq. (4.54) by solving the integral $\int \mathcal{N}\left(K_{\eta,u} K_{u,u}^{-1} \boldsymbol{u}, \mathbb{I}\nu \right) p\left(\boldsymbol{u}\right) \mathrm{d}\boldsymbol{u}$.

$$F_v(Z) = -\frac{1}{2} \log |Q_{\eta,\eta} + \nu\mathbb{I}| - \frac{1}{2} \boldsymbol{y}^T (Q_{\eta,\eta} + \nu\mathbb{I})^{-1} \boldsymbol{y} - \frac{N}{2} \log 2\pi - \frac{1}{2\nu} \mathrm{tr}(K_{\eta,\eta} - Q_{\eta,\eta}) \tag{4.54}$$

Finally the optimal 'free' distribution $\hat{\phi}(\boldsymbol{u})$ can be obtained by differentiating with respect to $\phi(\boldsymbol{u})$, and setting this to zero as shown in Eq. (4.55).

$$\hat{\phi}(\boldsymbol{u}) = \frac{\mathcal{N}\left(K_{\eta,u} K_{u,u}^{-1} \boldsymbol{u}, \mathbb{I}\nu \right) p\left(\boldsymbol{u}\right)}{\int \mathcal{N}\left(K_{\eta,u} K_{u,u}^{-1} \boldsymbol{u}, \mathbb{I}\nu \right) p\left(\boldsymbol{u}\right) \mathrm{d}\boldsymbol{u}} = \frac{z_c \mathcal{N}\left(\mu_v, \Sigma_v\right)}{z_c} \tag{4.55}$$

Where z_c and $\mathcal{N}\left(\boldsymbol{\mu}_v, \Sigma_v\right)$ are the constant and distribution from the product of two Gaussian distributions (see Appendix A.2) with mean and covariance; $\boldsymbol{\mu}_v = \nu^{-2} K_{u,u} A K_{u,\eta} \boldsymbol{y}$ and $\Sigma_v = K_{u,u} A K_{u,u}$ where $A = (K_{u,u} + \nu^{-2} K_{u,\eta} K_{\eta,u})^{-1}$.

Equation (4.54) is equivalent to that of the DTC approximation with the inclusion of a trace regularisation term. This means that the objective function in the optimisation is a true lower bound of the marginal likelihood. By substituting the optimal 'free'

distribution into Eq. (4.49) and using standard Gaussian conditionals the closed form predictive posterior can be obtained, as in Eq. (4.56).

$$q\left(\boldsymbol{\eta}_* \mid \boldsymbol{y}, \boldsymbol{\theta}\right) = \mathcal{N}\left(Q_{*,\eta}\tilde{K}_{\eta,\eta}^{-1}\boldsymbol{y}, K_{*,*} - Q_{*,\eta}\tilde{K}_{\eta,\eta}^{-1}Q_{\eta,*}\right) \tag{4.56}$$

Where $\tilde{K}_{\eta,\eta} = Q_{\eta,\eta} + \nu I$. The approximate posterior is identical to that of DTC meaning VFE can be thought of as DTC but penalised by a term proportional to the summed variances. This term protects against overfitting and forces the inducing inputs to better explain all the data, improving their optimised locations. The approach remains non-parametric because the inducing points become variational parameters meaning any additional inducing points will always increase the prediction quality, which cannot be claimed for model approximation methods.

An additional approach to approximating the posterior is to use a PEP framework [124]. The method seeks to approximate the joint-distribution in the form of Eq. (4.57).

$$p\left(\boldsymbol{\eta}_*, \boldsymbol{y}\right) = p\left(\boldsymbol{\eta}_* \mid \boldsymbol{y}\right)p\left(\boldsymbol{y}\right) \approx p\left(\boldsymbol{\eta}_*\right)\prod_n t_n(\boldsymbol{u}) = q^{un}\left(\boldsymbol{\eta}_*\right) \tag{4.57}$$

Where $(\cdot)^{un}$ indicates an unnormalised process. Equation (4.57) shows that only the likelihood term in the exact posterior is approximated and by a factor $t_n(\boldsymbol{u})$ — assumed to be Gaussian. PEP then iteratively modifies the factors in order to capture the behaviour the true likelihood imposes on the posterior, i.e. the best surrogate likelihood that approximates the posterior. The PEP algorithm involves three steps in which a fraction α of the approximate likelihood function is incorporated iteratively for each factor that needs to be approximated.

1. Deletion: a fraction of one approximate factor is removed in order to evaluate the cavity distribution (this is an approximate leave-one out joint, where $_{\backslash n}$ indicates leave-one out)[3], $q_{\backslash n}^{un}(\boldsymbol{\eta}_*) \propto q^{un}(\boldsymbol{\eta}_*)/t_n^{\alpha}(\boldsymbol{u})$.

2. Projection: a tilted distribution is projected onto the posterior distribution

[3] $p_{\backslash n}^{*}(\boldsymbol{\eta}_*) = p\left(\boldsymbol{\eta}_*, \boldsymbol{y}\right)/p\left(y_n \mid \eta_n\right) \approx q_{\backslash n}^{un}(\boldsymbol{\eta}_*) = q^{un}(\boldsymbol{\eta}_*)/t_n(\boldsymbol{u})$.

using the alpha-divergence[4] for unnormalised densities:

$$q^{un}_{\backslash n}(\boldsymbol{\eta}_*) \leftarrow \arg\min \overline{D}_\alpha(\tilde{p}(\boldsymbol{\eta}_*)||q^{un}_{\backslash n}(\boldsymbol{\eta}_*)).$$

The titled distribution is formulated by using the same fraction of the true likelihood as used in creating the cavity distribution, $\tilde{p}(\boldsymbol{\eta}_*) = q^{un}_{\backslash n}(\boldsymbol{\eta}_*)p^\alpha(y_n|\eta_n)$.

3. Update: An updated factor is calculated by the inclusion of a new fraction of the approximate factor, $t_n(\boldsymbol{u}) = t^{1-\alpha}_{n,old}(\boldsymbol{u})t^\alpha_{n,new}(\boldsymbol{u})$ where $t^\alpha_{n,new}(\boldsymbol{u}) = q^{un}(\boldsymbol{\eta}_*)/q^{un}_{\backslash n}(\boldsymbol{\eta}_*)$.

When a Gaussian likelihood is assumed the PEP approach has a closed form solution [124]. This is because the approximate factors can be defined at convergence as stable fixed points and the update step remains the same. The factor of the likelihood $t_n(\boldsymbol{u})$ and inducing variable distribution $q^{un}(\boldsymbol{u})$ are shown in Eqs. (4.58) and (4.59).

$$t_n(\boldsymbol{u}) = \mathcal{N}\left(K_{\eta_n,u}K^{-1}_{u,u}\boldsymbol{u}, \alpha D_{\eta_n,\eta_n} + \nu\right) \tag{4.58}$$

$$q^{un}(\boldsymbol{u}) = \mathcal{N}\left(K_{u,\eta}\bar{K}^{-1}_{\eta,\eta}\boldsymbol{y}, K_{u,u} - K_{u,\eta}\bar{K}^{-1}_{\eta,\eta}K_{u,\eta}\right) \tag{4.59}$$

Where $D_{\eta,\eta} = K_{\eta,\eta} - Q_{\eta,\eta}$. These lead to a closed form approximate log marginal likelihood $\log \mathcal{Z}_{PEP}$ and posterior $q(\boldsymbol{\eta}_* \,|\, \boldsymbol{y})$ defined in Eqs. (4.60) and (4.61) — where Eq. (4.61) is equivalent to the model approximation posterior.

$$\log \mathcal{Z}_{PEP} = -\frac{1}{2}\log|\bar{K}_{\eta,\eta}| - \frac{1}{2}\boldsymbol{y}^T\bar{K}^{-1}_{\eta,\eta}\boldsymbol{y} - \frac{N}{2}\log 2\pi - \frac{1-\alpha}{2\alpha}\sum_n \log(1 + \alpha D_{\eta_n,\eta_n}/\nu I) \tag{4.60}$$

$$q(\boldsymbol{\eta}_* \,|\, \boldsymbol{y}) = \mathcal{N}\left(Q_{*,\eta}\bar{K}^{-1}_{\eta,\eta}\boldsymbol{y}, K_{*,*} - Q_{*,\eta}\bar{K}^{-1}_{\eta,\eta}Q_{\eta,*}\right) \tag{4.61}$$

Interesting results occur when $\alpha = 1$ and as $\alpha \to 0$, the PEP posterior and log marginal likelihood become equivalent to the FITC and VFE approach respectively. This unifying view is helpful in understanding the effects of the parameter α. When

[4] An alpha-divergence is $D_\alpha(\mathbb{P}\,||\,\mathbb{Q}) = \frac{1}{\alpha(1-\alpha)}\int \alpha p(x) + (1-\alpha)q(x) - p(x)^\alpha q(x)^{1-\alpha}\mathrm{d}x$.

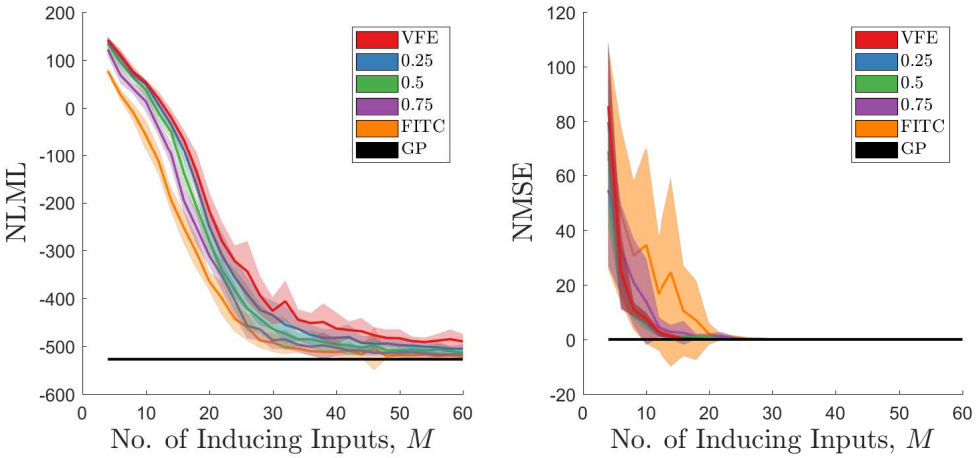


Figure 4.10: The effect of the number of inducing inputs M and α on the performance of the PEP formulation of sparse GPs averaged over 25 repeats for the NLML (left) and the NMSE (right). Shaded regions indicate $\pm\sigma$ confidence intervals.

$\alpha < 1$ the last term of the PEP log marginal likelihood $\frac{1-\alpha}{2\alpha}\sum_n \log(1 + \alpha D_{\eta_n,\eta_n}/\nu I)$ will act as a regularising term and making sure that the model generalises well to new outputs; the extreme of the penalty term being the VFE trace term. Bauer et al. produced an overview of the differences between the FITC and VFE approaches [126]. They state that FITC has several negative drawbacks, it can overestimate the marginal likelihood, underestimate the noise/nugget, is not guaranteed to improve when more inducing points are added and does not recover the true posterior. VFE in contrast, can overestimate the noise/nugget, does improve with more inducing points and will recover the true posterior where possible whilst providing a true lower bound of the marginal likelihood.

When employing a posterior approximation approach, the nugget term will need to be inferred as a hyperparameter, rather than a fixed term. This is because the nugget now includes a measure of the uncertainty introduced by using a low rank approximation when performing inference. It is noted that both VFE and PEP approximations result in a computational complexity of $\mathcal{O}(NM^2)$ for training with $\mathcal{O}(M)$ and $\mathcal{O}(M^2)$ for the mean and variance predictions [124, 127].

Figures 4.10 and 4.11 demonstrate the effect of additional inducing points and the α parameter for a different one-dimensional numerical example. Here the simulator output is a sample from a GP with zero mean and a SE covariance function; $\sigma_f^2 = 1$, $\omega = 30$. Sparse GPs models were created with $\alpha = 0$, 0.25, 0.5, 0.75, 1 and are compared to the full GP solution and training data in Fig. 4.11, where mean and $\pm 3\sigma$

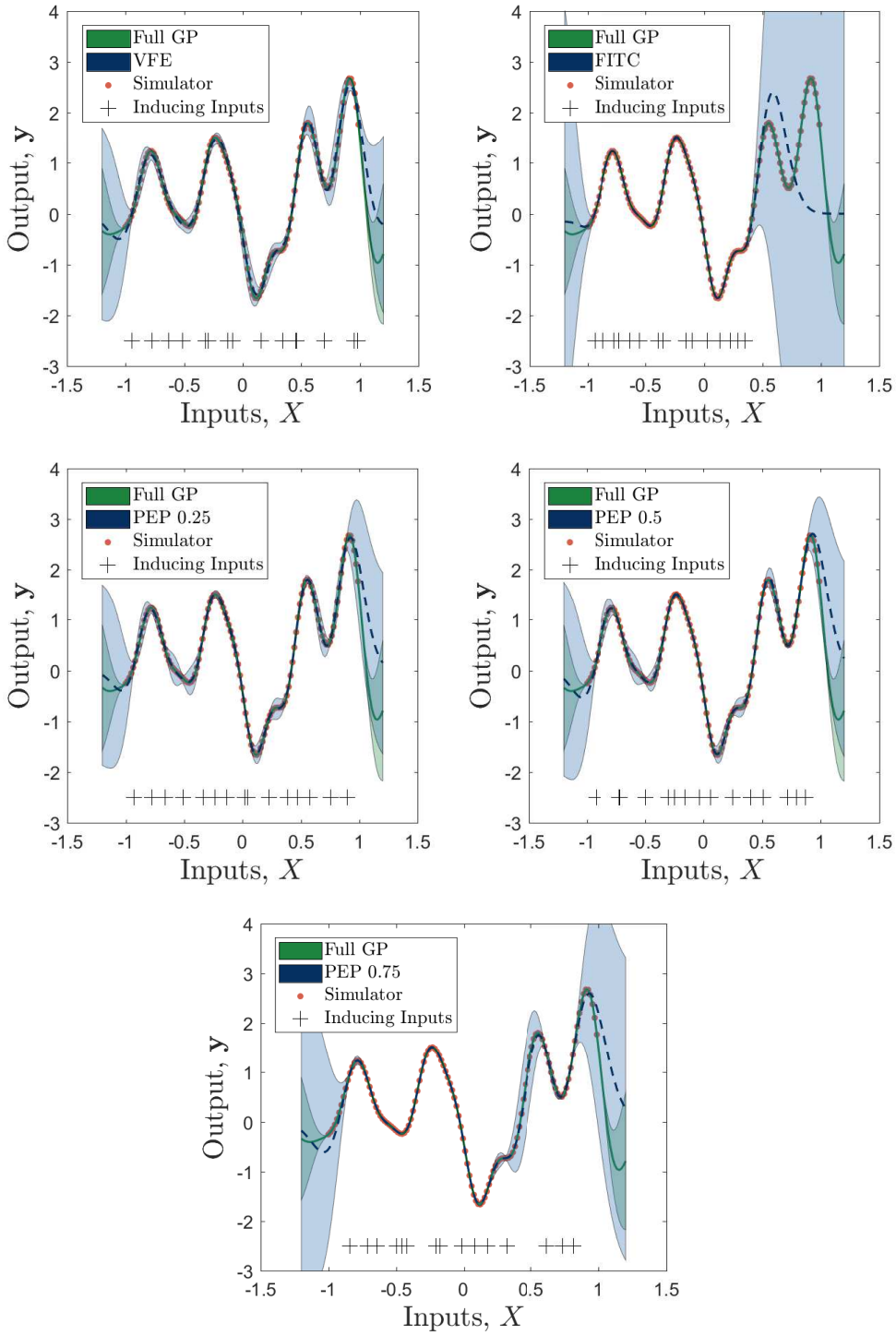


Figure 4.11: Predictions from posterior approximation and FITC sparse GPs with 15 inducing points, against a full GP, and training simulator data for a numerical example. Shaded regions indicate $\pm 3\sigma$ confidence levels. Top left panel, VFE ($\alpha = 0$); top right panel, FITC ($\alpha = 1$); middle left panel, PEP $\alpha = 0.25$; middle right panel, PEP $\alpha = 0.5$; and bottom panel PEP $\alpha = 0.75$.

confidence intervals are presented. A stochastic optimisation method was utilised for inferring the hyperparameters with 25 repeats to quantify the variance in the inference, presented in Fig. 4.10 as $\pm\sigma$ confidence intervals. It is demonstrated that the NLML ($-\log p(\boldsymbol{y} \mid X) \approx -F_v(Z) \approx -\log \mathcal{Z}_{PEP}$) reduces as more inducing points are added, stating that the model better explains the data given more inducing inputs. The NLML also increases with α, in contrast, the NMSE is high with larger variance for FITC, a clear indication that the method has experienced overfitting. It is noted that there is significant overlap in NMSE results for VFE and PEP when $\alpha = 0.25$, 0.5, indicating their predictions are very similar.

The PEP approach, when $\alpha = 0.25$, 0.5 provides better predictions of the data when compared to FITC and PEP at $\alpha = 0.75$, demonstrated by low NLMLs that correspond to low NMSEs. FITC and PEP when $\alpha = 0.75$, although showing low NLMLs, have high NMSEs with large variance, especially when the number of inducing points is low; which is a clear sign of overfitting. VFE tends to have high NLML with comparable NMSE to PEP when $\alpha = 0.25$, 0.5. Figure 4.11 demonstrates that the variance of the VFE prediction is larger than the full GP solution, with the variance of both the PEP formulations when $\alpha = 0.25$, 0.5, visually matching the full GP more closely. For these reasons it can be argued that PEP, when $\alpha = 0.25$, 0.5, preforms better in these examples. A close inspection of the NMSE in Fig. 4.10 for PEP when $\alpha = 0.5$ demonstrates lower values than any of the posterior approximation methods. This leads to the conclusion that PEP with an $\alpha = 0.5$ outperforms other α values (FITC and VFE included) which is consistent with the findings of Bui et al [124]. The question still arises of how to choose the α parameter. Optimisation is not advised as a value of 1 will lead to overfitting due to the FITC approximation. It is the experience of the author that a value of 0.5 should give satisfactory performance, in-keeping with the finding of Bui et al. [124].

4.3.3 Considerations for Sparse Gaussian Process Emulators

There are two main reasons why a sparse GP approximation can be useful in creating an emulator. Firstly, when a relatively large number of simulator runs are available, a sparse approximation can make inference practical. This is achieved by reducing the computational time complexity to $\mathcal{O}(NM^2)$ per simulator observation and reducing the memory requirement. Secondly, when predictions are required at a large number

of test inputs a moderate computational saving is made, $\mathcal{O}(M)$ and $\mathcal{O}(M^2)$ per test point. Applications of when these reasons may be applicable are presented in this section.

In the authors opinion, it is not commonplace that the simulator is run at a large number of parameter combinations. This problem mainly arises in a high dimensional parameter space where most of the parameters actively and significantly effect the output. Here even a space-filled design will result in a large number of simulator runs and a sparse GP approximation is applicable. Sparse GPs are more useful in a Bayesian optimisation [129, 130] or Bayesian history matching setting [71, 131]. Both methods often require predictions from the emulator for a large number of parameter combinations in order to accurately assess the output space for optimal solutions. The moderate computational saving in the prediction per test point means that a better exploration of the space can be performed. This becomes more important in a sequential design process as used in an entropy search or information gain approach [129]. These methods often predict based on a set grid size for the parameter space; reducing the computational load for prediction means a finer grid can be set. Due to the approximate nature of sparse GPs their use is not always needed or favourable for creating emulators. The approximation introduces a nugget term that cannot be fixed as it is a coupling between a noise parameter for the data and an estimation of the error introduced by a low rank approximation. This means that deterministic predictions at known simulator outputs are not possible, as is the case with the full GP emulator. This has to be considered when the code uncertainty affects the results of additional processes, as is the case with Bayesian optimisation and Bayesian history matching.

4.4 Extensions for Gaussian Processes

GPs have been adapted extensively throughout the literature for a variety of problems. Several GP technologies of note are outlined briefly. These include multivariate GP formulations, frameworks for predicting stochastic emulator outputs and techniques for incorporating dynamics.

4.4.1 Multivariate Gaussian Processes

Most applications involving the generation of an emulator will involve creating a surrogate of multiple simulator outputs i.e. $Y = \eta(X)$ where $X = \{\boldsymbol{x}_n\}_{n=1}^N$ of dimension D and $Y = \{\boldsymbol{y}_n\}_{n=1}^N$ of dimension k. However, GPs are univariate and therefore Eqs. (4.15) and (4.23) are formulated for a single output. The most naive approach is to construct multiple independent univariate GP emulators for each output. This may be valid in situations where the outputs are uncorrelated and there is no dependence defined within the computer code between inputs and outputs. Nonetheless, in most emulation contexts it is expected that there will be joint correlation between the inputs and outputs as well as dependence between outputs. This means that an independent GP prior assumption would result in a loss of information. Outputs are often different quantities which means there is no need for a cross-mean function term. As a consequence the adaptation from a univariate to multivariate GP is mainly concerned with specifying a GP prior that captures the cross-dependences in the covariance matrix, with a general multi-output GP form defined in Eq. (4.62).

$$Y \sim \mathcal{GP}\left((\mathbb{I}_k \otimes H_\eta)\boldsymbol{\beta}\,, V_{\eta,\eta}\right) \tag{4.62}$$

Where H_η is the design matrix of p basis functions, $\boldsymbol{\beta}$ is a vector of kp coefficients and $V_{\eta,\eta}$ is the covariance matrix. Formulations of multivariate GPs are broadly categorised by whether the covariance matrix is separable or non-separable.

A separable covariance matrix assumes that $V_{\eta,\eta} = \Sigma\, c(X, X')$ which is equivalent to $V_{\eta,\eta} = \Sigma \otimes A_{\eta,\eta}$, where Σ is a hyperparameter [104, 132]. This structure keeps the problem tractable and allows Σ to be marginalised out using a non-informative Jeffreys prior. The problem with a separable approach is that only one covariance function can be specified, this must be applicable to all outputs in the model. Additionally, this separability means that the covariance between two outputs is zero, meaning that observing one output will not provide information about any other output (a kind of Markov property) [107].

A non-separable approach alleviates these problems allowing different covariance functions for each output. Two approaches are convolution and coregionalisation methods [107]. The convolution approach treats GPs as outputs of stable linear

filters, where a GP is the same as convolving a smoothing kernel with a white noise process [107, 133–135]. A covariance matrix with cross-dependent terms and individual covariance functions for each output can be defined by summing multiple convolutions. Per contra, the linear model of coregrionalisation sums linear combinations of a number of independent GP models in order to create a multiple output GP [107].

An alternative approach to the form in Eq. (4.62) is a deep GP [136]. The simplest deep GP is for a single output where two GP models are connected, i.e. the output of the first GP is the input to the next, where the latent function space is marginalised out using variational inference (retaining the Bayesian Occam's razor). The potential hierarchy of combined GPs is deemed flexible enough to produce multiple-output predictions [136].

4.4.2 Stochastic Emulators

The simulators considered so far have been deterministic, however certain computer models may also be stochastic, such as stochastic FEA [137]. In this context, a GP emulator must have a mechanism for accurately capturing the heteroscedastic behaviour. Broadly, GP technologies for predicting heteroscedastic processes involve two GPs; one for the mean and one for the variance. Andrianakis et al. emulate the mean and variance as two distinctly separate GPs, incorporating the prediction of each within a Bayesian History Matching (BHM) setting [109]. Another approach by Lázaro-Gredilla and Titsias is to combine two GPs, with the second being introduced as an exponentiated noise model. This formulation is no longer tractable, and leads to the definition of a variational approximation [138].

4.4.3 Dynamical Gaussian Processes

When the simulator is predicting dynamic outputs, i.e. time histories a of particular quantity, it may be beneficial to incorporate temporal knowledge into the GP emulator. There are several approaches that exist within the literature, categorised broadly into two main approaches: Autoregressive (AR) and state-space formulations. An AR approach models the next output in a times series as some mapping from past observations, whereas a state-space framework describes the outputs as a Markov

process with one or more states evolving in time by means of a transition function. Frigola-Alcalde provides a detailed overview in [139].

The AR approach leads to the formulation of a Nonlinear Auto-Regressive eXogenous inputs (NARX) (where exogenous are external inputs) model, where previous outputs of the model become inputs to the next time point. In the GP formulation these past observations become part of the input set to a GP, mapping the nonlinear transition to the next time point [139–142].

A nonlinear state-space approach can be formulated in a variety of ways [139, 143, 144]. Firstly, either the transition or observational models or both can be assumed a GP. These prior assumptions are used to reflect the belief that the nonlinearity is contained within either the system dynamics or the measurement respectively [139]. Additionally both the transition and observational models can be GPs (known as the full GP-state space model), however this can lead to non-identifiability problems, as both models are flexible nonlinear functions [139, 144]. Key challenges to this approach are computational complexity, identifiability issues and interpretability.

4.5 Conclusion

Simulators are used throughout engineering and are an integral part of forward model-driven SHM. The majority of statistical methods and optimisation techniques that analyse or incorporate simulators require numerous evaluations. These methods may not be practically feasible when the simulator is computationally expensive to run. For this reason emulators, computationally efficient surrogates of a simulator, are employed.

A variety of tools have been implemented as emulators throughout the literature, notably ANNs, PCE, BLA and GPs. It has been discussed that only BLA and GPs quantify the uncertainty associated with replacing the simulator with an emulator — known as code uncertainty. Moreover, both ANNs and PCE can overfit, and without providing code uncertainty the user is unaware when this occurs. BLA is an approximation of Bayesian inference and only considers the mean and variance. In contrast GPs have closed form solutions to Bayesian inference, when the function can be assumed Gaussian distributed. For these reasons, when the outputs are considered jointly Gaussian, a GP emulator will be the most rigorous form surrogate model,

and therefore is utilised in this thesis.

The chapter has outlined derivations of a GP for the purpose of emulating a deterministic simulator, along with methods for dealing with numerical issues associated with the 'noise-free' assumption. In addition, diagnostics have been implemented on a numerical example presenting a framework for validating an emulator. An emulator must be constructed from a finite set of simulator runs, and a GMLHD has been demonstrated to improve GP predictions.

When the number of input variables N is large, GPs can become numerically intractable as they rely on the inversion of an $N \times N$ matrix which has a time complexity of $\mathcal{O}(N^3)$. Sparse GP methods have been proposed to reduce the time complexity to $\mathcal{O}(NM^2)$ and considerations for implementation in an emulator context have been outlined. Finally, other GP extensions within the literature have been presented, such as multivariate, heteroscedastic and dynamical GPs.

Chapter 5

Bayesian Calibration and Bias Correction

In the previous chapter model discrepancy, that occurs due to model form errors, was outlined as a problem in generating predictions from simulators that accurately represent real world observations. This is a particular issue for forward model-driven SHM as a key objective is to generate statistically representative outputs of observational damage states from a simulator. This means that the calibration procedures implemented in forward model-driven SHM must consider model discrepancy as a source of uncertainty. Bayesian Calibration and Bias Correction (BCBC) is one such approach, seeking to calibrate simulator parameters whilst inferring the model discrepancy functional distributions.

The following chapter begins with a discussion of the literature before outlining the BCBC methodology. Subsequent sections demonstrate the technique on two case studies; a three story and a five storey building structure, providing a discussion on the benefits and challenges with the formulation. Lastly, conclusions are presented outlining the methodologies effectiveness within a forward model-driven framework.

5.1 Literature Review

BCBC (also known as the 'Kennedy and O'Hagan approach' or a modular Bayesian technique) was developed in 2001 by Kennedy and O'Hagan [68] as part of a discussion

about the correct procedure for calibrating deterministic simulators in a Bayesian manner. The key development of the paper was to outline the sources of uncertainty within a computer simulation, highlighting that model discrepancy should be inferred, along with parameter uncertainties, and proposing that it could be modelled using a GP prior. Their proposed statistical model for calibrating a simulator is as defined in Eq. (5.1).

$$z(\boldsymbol{x}) = \zeta(\boldsymbol{x}) + e = \rho\eta(\boldsymbol{x}, \boldsymbol{\theta}) + \delta(\boldsymbol{x}) + e \tag{5.1}$$

This statistical model provides a belief about the relationship between observations $z(\boldsymbol{x})$ and simulator $\eta(\boldsymbol{x}, \boldsymbol{\theta})$ that depends on a set of inputs $\boldsymbol{x}$ and parameters $\boldsymbol{\theta}$. The model assumes that the combination of simulator and model discrepancy $\delta(\boldsymbol{x})$ are equivalent to the 'true' process $\zeta(\boldsymbol{x})$, where model discrepancy is assumed to have a functional form. In the original formulation a regression parameter ρ is used to weight the evidence provided by the simulator relative to the model discrepancy; with this parameter informing the relative weighting between the model discrepancy and simulator — although some more recent formulations remove this term. Lastly, the observational data $z(\boldsymbol{x})$ is modelled as the 'true' process with the addition of independent observational uncertainty (a Gaussian homoscedastic noise).

The framework has been applied and adapted several times within engineering. Bayarri et al. implemented the methodology on a spot weld FEA model where they discuss the differences between a modular Bayesian approach and full Bayesian analysis, stating similarities in the results [69]. Higdon et al. proposed a multivariate formulation using principle components modelled as GPs [103]. The method was demonstrated on a simulator modelling implosion in a cylinder, where output predictions were demonstrated to fit the data well. However, comments in the paper indicate non-identifiability issues between the parameters and model discrepancy as well as problems in scenarios where the simulator cannot be modelled as a standard GP. The framework's approach to model discrepancy is also discussed in a general review of model updating [16] without any definitive conclusions. In engineering design, Arendt et al. present an application of the univariate method clearly indicating the problems associated with non-identifiability between the parameters and model discrepancy when non-informative or inadequate priors are used [105]. The issues with non-identifiability are approached again by Arendt et al. where multivariate GPs with separable covariances ([104]) are incorporated in order to better define

the calibration space [132]. Nonetheless, the method did not completely solve these problems, mainly due to numerical instabilities in the multivariate GPs. In addition, Arendt et al. use a preposterior technique, where the GP covariance was estimated prior to calibration using a least squares technique, in an attempt to improve the non-identifiability issues [145]. Finally, Brynjarsdóttir and O'Hagan discussed the importance of inferring model discrepancy and the problem of inappropriate model discrepancy prior specification [146]. They state that the prior distribution for the parameters must be informative where possible and the GP prior constrained to reflect prior physical knowledge.

The literature clearly discusses the issues associated with parameter inference when a naive GP prior is formed for the model discrepancy or uninformative priors are used for the parameters. This is especially problematic in a model updating view of SHM where the updated parameters are used to make inferences about the structure's health state. It is also problematic when extrapolation is required, resulting from incorrect inference of the simulator parameters. On the other hand in a forward model-driven SHM context often only interpolation of the outputs is required. This may be possible with the BCBC framework when an unconstrained GP prior and informative parameter priors are utilised, as the inferred parameters and model discrepancy will be fitted to the training input domain.

5.2 Methodology

BCBC aims to calibrate the statistical model of the form described in Eq. (5.1) using Bayesian inference, i.e. $p(\boldsymbol{\theta} \,|\, \boldsymbol{d}) \propto p(\boldsymbol{d} \,|\, \boldsymbol{\theta}) p(\boldsymbol{\theta})$, where $\boldsymbol{d} = (\boldsymbol{y}^{\mathsf{T}}, \boldsymbol{z}^{\mathsf{T}})^{\mathsf{T}}$ is a combined data set of simulator and observational outputs. The data are obtained from a finite set of N simulator evaluations $D_y = \{(\boldsymbol{x}_1, \boldsymbol{t}_1), \ldots, (\boldsymbol{x}_N, \boldsymbol{t}_N)\}$ (where $\boldsymbol{t}$ are potential parameters where $\boldsymbol{\theta}$ may be contained) and their corresponding outputs $\boldsymbol{y}$, in addition to a finite set of n observations $\boldsymbol{z}$ obtained at $D_z = \{\boldsymbol{x}_1^z, \ldots, \boldsymbol{x}_n^z\}$ (these can be different locations to the simulator inputs). It is common that $N \gg n$ as simulator evaluations are often easier to obtain than experimental data.

The likelihood function is assumed Gaussian, i.e. $p(\boldsymbol{d} \,|\, \boldsymbol{\theta}, \boldsymbol{\phi}) = \mathcal{N}(\boldsymbol{m}_d(\boldsymbol{\theta}), \boldsymbol{V}_d(\boldsymbol{\theta}))$, and is constructed from GP models (using Eq. (5.1)) dependent on the hyperparameters $\boldsymbol{\phi}$. Specifically the approach assumes that both the simulator $\eta(\boldsymbol{x}, \boldsymbol{\theta})$ and model discrepancy $\delta(\boldsymbol{x})$ can be modelled independently by GPs, defined in Eqs. (5.2)

and (5.3) respectively — where the simulator output $\boldsymbol{y} = \eta(\boldsymbol{x}, \boldsymbol{\theta})$.

$$\boldsymbol{y} = \eta(\boldsymbol{x}, \boldsymbol{\theta}) \sim \mathcal{GP}\left(m_\eta(\boldsymbol{x}, \boldsymbol{\theta}), k_\eta((\boldsymbol{x}, \boldsymbol{\theta}), (\boldsymbol{x}', \boldsymbol{\theta}'))\right) \tag{5.2}$$

$$\delta(\boldsymbol{x}) \sim \mathcal{GP}\left(m_\delta(\boldsymbol{x}), k_\delta(\boldsymbol{x}, \boldsymbol{x}')\right) \tag{5.3}$$

The simulator, modelled as a 'noise-free' GP emulator, is specified by a mean $m_\eta(\cdot)$ and covariance function $k_\eta(\cdot, \cdot)$ with hyperparameters $\boldsymbol{\phi}_\eta$ and seeks to emulate over the input $\boldsymbol{x}$ and parameter $\boldsymbol{\theta}$ space. The prior belief for the model discrepancy GP is described by the mean $m_\delta(\cdot)$ and covariance function $k_\delta(\cdot, \cdot)$ with hyperparameters $\boldsymbol{\phi}_\delta$ and describes the model discrepancy when the 'true' calibrated parameters $\boldsymbol{\theta}$ are known.

The prior model for the observational outputs $\boldsymbol{z}$ are constructed using Eq. (5.1). The statistical model assumes independent, normally distributed observational uncertainties $\mathcal{N}(\boldsymbol{0}, \sigma_n^2)$. The three components form the observational output prior defined in Eq. (5.4).

$$\boldsymbol{z} \sim \mathcal{GP}\left(\rho m_\eta(\boldsymbol{x}, \boldsymbol{\theta}) + m_\delta(\boldsymbol{x}), \rho^2 k_\eta((\boldsymbol{x}, \boldsymbol{\theta}), (\boldsymbol{x}', \boldsymbol{\theta}')) + k_\delta(\boldsymbol{x}, \boldsymbol{x}') + \mathbb{I}\sigma_n^2\right) \tag{5.4}$$

Which is dependent on the hyperparameters $\boldsymbol{\phi} = \{\boldsymbol{\phi}_\eta, \boldsymbol{\phi}_\delta, \sigma_n^2, \rho\}$. The joint Gaussian likelihood $p(\boldsymbol{d} \,|\, \boldsymbol{\theta}, \boldsymbol{\phi})$ is then formed from the mean and covariance presented in Eqs. (5.5) and (5.6).

$$\mathbb{E}(\boldsymbol{d} \,|\, \boldsymbol{\theta}, \boldsymbol{\phi}) = \boldsymbol{m}_d(\boldsymbol{\theta}) = H(\boldsymbol{\theta})\boldsymbol{\beta} = \begin{bmatrix} H_\eta(D_y) & \boldsymbol{0} \\ \rho H_\eta(D_z(\boldsymbol{\theta})) & H_\delta(D_z) \end{bmatrix} \boldsymbol{\beta} \tag{5.5}$$

$$\mathbb{V}(\boldsymbol{d} \,|\, \boldsymbol{\theta}, \boldsymbol{\phi}) = \boldsymbol{V}_d(\boldsymbol{\theta}) = \begin{bmatrix} K_\eta(D_y) & \rho K_\eta(D_y, D_z(\boldsymbol{\theta}))^\mathsf{T} \\ \rho K_\eta(D_y, D_z(\boldsymbol{\theta})) & \rho^2 K_\eta(D_z(\boldsymbol{\theta})) + K_\delta(D_z) + \mathbb{I}_n \sigma_n^2 \end{bmatrix} \tag{5.6}$$

Where $D_z(\boldsymbol{\theta})$ are the observation inputs D_z augmented by the 'true' calibrated parameter i.e. $D_z(\boldsymbol{\theta}) = \{(\boldsymbol{x}_1^z, \boldsymbol{\theta}), \ldots, (\boldsymbol{x}_n^z, \boldsymbol{\theta})\}$; required for the evaluation of the

emulator's covariance function[1]. The identity matrix $\mathbb{I}_n$ is $n \times n$ and $\boldsymbol{\beta}$ are hyperparameters of the mean function (the hyperparameters $\boldsymbol{\beta}$ are part of the sets $\boldsymbol{\phi}_\eta$ and $\boldsymbol{\phi}_\delta$).

The hyperparameters $\boldsymbol{\phi} = \{\boldsymbol{\phi}_\eta, \boldsymbol{\phi}_\delta, \sigma_n^2, \rho\}$ for the statistical model in Eqs. (5.5) and (5.6) are inferred either in a fully Bayesian manner or using a plug-in approach — for reasons discussed in Section 4.2 a plug-in (empirical Bayes) method is implemented. Within the literature it is common to approach inference of the hyperparameters $\boldsymbol{\phi}$ in a modular manner (the reader is referred to [68, 69] for a more in-depth discussion) whereby Eq. (5.2) is fitted to a set of simulator runs to infer plug-in estimates of the hyperparameters $\hat{\boldsymbol{\phi}}_\eta$. These fixed plug-in estimates of $\hat{\boldsymbol{\phi}}_\eta$ are incorporated into the conditional distribution $p(\boldsymbol{z} \,|\, \boldsymbol{y}, \boldsymbol{\phi})$ and the remaining hyperparameters $\{\boldsymbol{\phi}_\delta, \sigma_n^2, \rho\}$ are inferred. The statistical model with the plug-in estimates $\hat{\boldsymbol{\phi}}$ — making it an empirical Bayes approach — are subsequently utilised in Bayesian inference of the parameters $\boldsymbol{\theta}$. Predictive posterior distributions of the output quantity can be inferred using the parameter posterior distribution.

The modular approach is summarised as follows with the consecutive sections providing more detail on each stage:

- The simulator is run for a finite set of N inputs $\boldsymbol{x}$ (the same $\forall\ \boldsymbol{\theta}$) and parameters $\boldsymbol{\theta}$ to obtain the outputs $\boldsymbol{y}$. The plug-in estimates of the hyperparameters $\hat{\boldsymbol{\phi}}_\eta$ are inferred for the GP emulator prior in Eq. (5.2).

- Observational outputs $\boldsymbol{z}$ are obtained for a finite set of n inputs $\boldsymbol{x}_z$ where typically $n \ll N$. The plug-in estimates of the hyperparameters $\{\boldsymbol{\phi}_\delta, \sigma_n^2, \rho\}$ are inferred for the model discrepancy GP. This GP maps from the emulator output — using $\hat{\boldsymbol{\phi}}_\eta$ and with $\boldsymbol{\theta}$ marginalised out — to the experimental data.

- The posterior distribution for the parameters $\boldsymbol{\theta}$ are inferred using Bayesian calibration i.e. $p(\boldsymbol{\theta} \,|\, \boldsymbol{d}, \boldsymbol{\phi}) \propto p(\boldsymbol{d} \,|\, \boldsymbol{\theta}, \boldsymbol{\phi}) p(\boldsymbol{\theta}) p(\boldsymbol{\phi})$. Full Bayesian analysis would require integrating out $\boldsymbol{\phi}$, however as this is intractable. The posterior $p(\boldsymbol{\theta} \,|\, \boldsymbol{d}, \boldsymbol{\phi})$ is therefore conditioned using the plug-in estimates $\hat{\boldsymbol{\phi}}$ (an empirical Bayes approach).

- The unconditional predictive posterior distribution of the observations $p(\boldsymbol{z} \,|\, \boldsymbol{d}, \boldsymbol{\phi})$ is generated by integrating out the inferred posterior parameter distribution $p(\boldsymbol{\theta} \,|\, \boldsymbol{d}, \boldsymbol{\phi})$.

[1]The notation $K_\eta(D_z(\boldsymbol{\theta}))$ relates to the covariance function $k_\eta\left((\boldsymbol{x}^z, \boldsymbol{\theta}), (\boldsymbol{x}^{z\prime}, \boldsymbol{\theta}')\right)$.

5.2.1 Emulator inference

The first stage of the modular BCBC approach is to infer the plug-in estimates of the hyperparameters $\boldsymbol{\phi}_\eta$. This is performed by fitting a GP emulator (Eq. (5.2)) to map the relationship between a finite set of N inputs $\boldsymbol{x}$ and parameters $\boldsymbol{t}$ — collectively referred to as $D_y = \{(\boldsymbol{x}_1, \boldsymbol{t}_1), \ldots, (\boldsymbol{x}_N, \boldsymbol{t}_N)\}$ — to the simulator outputs $\boldsymbol{y}$.

The complete specification of the GP prior in Eq. (5.2) requires the definition of a mean and covariance function. The choice must reflect prior assumptions about the functional form of the simulator (see Section 4.2 for more information). The mean function $m_\eta(\boldsymbol{x}, \boldsymbol{t})$ can be formed by any parametric basis functions that are linear in the coefficients $\boldsymbol{\beta}_\eta$ as demonstrated in Eq. (5.7).

$$m_\eta(\boldsymbol{x}, \boldsymbol{t}) = H_\eta(D_y)\boldsymbol{\beta}_\eta \tag{5.7}$$

Generally a constant mean function is used, i.e. $H_\eta(D_y) = 1$ unless prior information about the simulator function is known. In addition, the choice of $H_\eta(D_y)$ is restricted for a closed form solution to BCBC, where the expectation with respect to parameter prior $p(\boldsymbol{\theta})$ must be tractable.

The covariance function $k_\eta((\boldsymbol{x}, \boldsymbol{t}), (\boldsymbol{x}', \boldsymbol{t}'))$ is chosen to reflect the prior smoothness of the simulator function. As discussed in Section 4.2, an SE covariance function is appropriate when the simulator output can be considered smooth, as assumed here. This choice of covariance function also means that the expectation in relation to a Gaussian prior for $p(\boldsymbol{\theta})$ will remain tractable, leading to a closed form solution for marginalising the parameters from the emulator GP — required for the model discrepancy inference stage. In addition a separable covariance structure is implemented for the inputs $\boldsymbol{x}$ and parameters $\boldsymbol{t}$. The separate covariance functions are combined using a product (equivalent to a logical 'AND' statement) which reflects their dependence. The prior ARD covariance function is presented in Eq. (5.8).

$$k_\eta((\boldsymbol{x}, \boldsymbol{\theta}), (\boldsymbol{x}', \boldsymbol{\theta}')) = \sigma_\eta^2 \exp\left(-(\boldsymbol{x} - \boldsymbol{x}')^\mathsf{T}\Omega_x(\boldsymbol{x} - \boldsymbol{x}')\right) \exp\left(-(\boldsymbol{t} - \boldsymbol{t}')^\mathsf{T}\Omega_t(\boldsymbol{t} - \boldsymbol{t}')\right) \tag{5.8}$$

Where σ_η^2 is the scale factor hyperparameter, Ω_x and Ω_t are diagonal matrices of rough-

ness parameters for each dimension d, grouped $\boldsymbol{\psi}_\eta = \{\Omega_x, \Omega_t\}$. The computation time in generating the covariance matrix can be increased by using the kronecker product rules. This means that $K_\eta = \sigma_\eta^2 A_t \otimes A_x$ (where $A_t = \exp\left(-(\boldsymbol{t}-\boldsymbol{t}')^\mathsf{T}\Omega_t(\boldsymbol{t}-\boldsymbol{t}')\right)$ and $A_x = \exp\left(-(\boldsymbol{x}-\boldsymbol{x}')^\mathsf{T}\Omega_x(\boldsymbol{x}-\boldsymbol{x}')\right)$, i.e. the corresponding correlation matrices). This results in computational savings when calculating $K_\eta^{-1} = \sigma_\eta^{-2} A_t^{-1} \otimes A_x^{-1}$ and $|K_\eta| = \sigma_\eta^{2n_t n_x} |A_t|^{n_x} \otimes |A_x|^{n_t}$ where n_t and n_x are the number of parameters and inputs respectively (i.e. $N = n_t n_x$).

The objective of this stage is to infer the plug-in estimates $\hat{\boldsymbol{\phi}}_\eta = \{\hat{\boldsymbol{\beta}}_\eta, \hat{\sigma}_\eta^2, \hat{\boldsymbol{\psi}}_\eta\}$. Using the same weak prior approach as in Section 4.2, both $\boldsymbol{\beta}_\eta$ and σ_η^2 can be marginalised out, leaving the MLE estimation of $\boldsymbol{\psi}_\eta$ via optimising the NLML $-\log p\left(\boldsymbol{\psi}_\eta \,|\, \boldsymbol{y}\right)$ (see Section 4.2 for full mathematical definitions and reasoning behind not marginalising out $\boldsymbol{\psi}_\eta$). The posterior predictions of the GP emulator $p\left(\boldsymbol{\eta} \,|\, \boldsymbol{y}, \boldsymbol{\psi}_\eta\right)$ can be obtained using Gaussian conditionals as demonstrated in Section 4.2; this does not need to be performed for BCBC and serves only for visualisation and emulator diagnostic purposes.

5.2.2 Model Discrepancy and Observational Uncertainty Inference

The second stage of the modular BCBC approach is to infer the plug-in estimates of the hyperparameters $\{\boldsymbol{\phi}_\delta, \sigma_n^2, \rho\}$. These need to be inferred independently of the parameter set $\boldsymbol{\theta}$. To do this the emulator hyperparameters $\hat{\boldsymbol{\phi}}_\eta$ are fixed and predictions independent of $\boldsymbol{\theta}$ obtained by marginalising out $\boldsymbol{\theta}$ by conditioning on the prior $p(\boldsymbol{\theta})$. This leads to a residual between the observational data and the uncertain emulator predictions, independent of $\boldsymbol{\theta}$, with which the model discrepancy GP is inferred.

To fully specify the observational GP model a mean and covariance function must be defined for the model discrepancy GP. The choice of functions are more flexible than with the emulator, as they do not have to be integrated with respect to $p(\boldsymbol{\theta})$. The mean and covariance functions are assumed to have the form stated in Eqs. (5.9) and (5.10).

$$m_\delta(\boldsymbol{x}^z) = H_\delta(D_z)\boldsymbol{\beta}_\delta \tag{5.9}$$

$$k_\delta(\boldsymbol{x}^z, \boldsymbol{x}^{z\prime}) = \sigma_\delta^2 c(\boldsymbol{x}^z, \boldsymbol{x}^{z\prime}; \boldsymbol{\psi}_\delta) \tag{5.10}$$

Where the model discrepancy hyperparameter set is $\boldsymbol{\phi}_\delta = \{\boldsymbol{\beta}_\delta, \sigma_\delta^2, \boldsymbol{\psi}_\delta\}$; the mean function coefficient, scale factor and correlation hyperparameters for the model discrepancy GP.

In order to perform plug-in inference of the hyperparameters $\{\boldsymbol{\phi}_\delta, \sigma_n^2, \rho\}$ the equation $p\left(\boldsymbol{\phi}_\delta, \sigma_n^2, \rho \,|\, \boldsymbol{d}, \boldsymbol{\phi}_\eta\right) \propto p\left(\boldsymbol{z} \,|\, \boldsymbol{y}, \boldsymbol{\phi}\right) p\left(\boldsymbol{\phi}_\delta, \sigma_n^2, \rho\right)$ should be formed and maximised. The distribution $p\left(\boldsymbol{z} \,|\, \boldsymbol{y}, \boldsymbol{\phi}\right)$ cannot be calculated analytically, but $p\left(\boldsymbol{z} \,|\, \boldsymbol{y}, \boldsymbol{\phi}, \boldsymbol{\theta}\right)$ is known and normally distributed. By integrating out $\boldsymbol{\theta}$ from the first and second moments of $p\left(\boldsymbol{z} \,|\, \boldsymbol{y}, \boldsymbol{\phi}, \boldsymbol{\theta}\right)$ (i.e. moment matching) an approximation of the distribution $p\left(\boldsymbol{z} \,|\, \boldsymbol{y}, \boldsymbol{\phi}\right)$ is obtained and utilised for inference of $\{\boldsymbol{\phi}_\delta, \sigma_n^2, \rho\}$. The marginalisation of $\boldsymbol{\theta}$ from the conditional mean function is presented in Eq. (5.11).

$$\mathbb{E}\left(\boldsymbol{z} \,|\, \boldsymbol{y}, \boldsymbol{\phi}\right) = \int \mathbb{E}\left(\boldsymbol{z} \,|\, \boldsymbol{y}, \boldsymbol{\phi}, \boldsymbol{\theta}\right) p\left(\boldsymbol{\theta}\right) \mathrm{d}\boldsymbol{\theta} = H_\delta \boldsymbol{\beta}_\delta + \rho \mathbb{E}_\theta\left(\boldsymbol{y} \,|\, \boldsymbol{\phi}_\eta\right) \tag{5.11}$$

This is prior model discrepancy mean in addition to the posterior emulator prediction unconditioned on $\boldsymbol{\theta}$, where $\mathbb{E}_\theta\left(\boldsymbol{y} \,|\, \boldsymbol{\phi}_\eta\right) = \hat{\eta}(D_z)$ — the expectation of the emulator mean with respect to the prior on $\boldsymbol{\theta}$. The ith element of the mean vector is calculated as shown in Eq. (5.12).

$$\begin{aligned}\mathbb{E}\left(z_i \,|\, \boldsymbol{y}, \boldsymbol{\phi}\right) = H_\delta(D_{z,i})\boldsymbol{\beta}_\delta + \rho\left(\int H_\eta(D_{z,i}(\boldsymbol{\theta}))p\left(\boldsymbol{\theta}\right)\mathrm{d}\boldsymbol{\theta}\right)\hat{\boldsymbol{\beta}}_\eta \\ + \rho\left(K_\eta\left(D_{z,i}(\boldsymbol{\theta}), D_{y,j}\right)^\mathsf{T} p\left(\boldsymbol{\theta}\right)\mathrm{d}\boldsymbol{\theta}\right) K_\eta(D_y)^{-1}(\boldsymbol{y} - H(D_y)\hat{\boldsymbol{\beta}}_\eta)\end{aligned} \tag{5.12}$$

Where $i = 1, \ldots, n$ and $j = 1, \ldots, N$. In the same manner $\boldsymbol{\theta}$ is marginalised out of the conditional covariance, Eq. (5.13).

$$\mathbb{V}\left(\boldsymbol{z} \,|\, \boldsymbol{y}, \boldsymbol{\phi}\right) = \int \mathbb{V}\left(\boldsymbol{z} \,|\, \boldsymbol{y}, \boldsymbol{\phi}, \boldsymbol{\theta}\right) p\left(\boldsymbol{\theta}\right) \mathrm{d}\boldsymbol{\theta} = \mathbb{I}_n \sigma_n^2 + K_\delta(D_z) + \rho^2 \int \mathbb{V}\left(\boldsymbol{y} \,|\, \boldsymbol{\theta}, \boldsymbol{\phi}_\eta\right) \mathrm{d}\boldsymbol{\theta} \tag{5.13}$$

This is the prior model discrepancy and observational uncertainty covariance summed

with the unconditional emulator posterior covariance, where $\int \mathbb{V}\left(\boldsymbol{y} \,|\, \boldsymbol{\theta}, \boldsymbol{\phi}_\eta\right) \mathrm{d}\boldsymbol{\theta} = C$ and $V = \mathbb{I}_n \sigma_n^2 + K_\delta(D_z) + \rho^2 C$. The ith, jth $(n \times n)$ element of C is shown in Eq. (5.14).

$$
\begin{aligned}
C_{i,j} = & \int K_\eta(D_{z,i}(\boldsymbol{\theta}), D_{z,j}(\boldsymbol{\theta})) p\left(\boldsymbol{\theta}\right) \mathrm{d}\boldsymbol{\theta} \\
& - \operatorname{tr}\left(K_\eta(D_\eta)^{-1} \int K_\eta\left(D_{z,j}(\boldsymbol{\theta}), D_{y,k}\right) K_\eta\left(D_{z,i}(\boldsymbol{\theta}), D_{y,l}\right)^{\mathsf{T}} p\left(\boldsymbol{\theta}\right) \mathrm{d}\boldsymbol{\theta} \right) \\
& + \operatorname{tr}\left(W_\eta(D_\eta) \int H_\eta\left(D_{z,j}(\boldsymbol{\theta})\right) H_\eta\left(D_{z,i}(\boldsymbol{\theta})\right)^{\mathsf{T}} p\left(\boldsymbol{\theta}\right) \mathrm{d}\boldsymbol{\theta} \right) \\
& - \operatorname{tr}\left(W_\eta(D_\eta) H_\eta(D_\eta)^{\mathsf{T}} K_\eta(D_\eta)^{-1} \int K_\eta\left(D_{z,j}(\boldsymbol{\theta}), D_{y,k}\right) H_\eta\left(D_{z,i}(\boldsymbol{\theta})\right)^{\mathsf{T}} p\left(\boldsymbol{\theta}\right) \mathrm{d}\boldsymbol{\theta} \right) \\
& - \operatorname{tr}\left(K_\eta(D_\eta)^{-1} H_\eta(D_\eta) W_\eta(D_\eta) \int H_\eta\left(D_{z,j}(\boldsymbol{\theta})\right) K_\eta\left(D_{z,i}(\boldsymbol{\theta}), D_{y,l}\right)^{\mathsf{T}} p\left(\boldsymbol{\theta}\right) \mathrm{d}\boldsymbol{\theta} \right) \\
& + \operatorname{tr}\left(K_\eta(D_\eta)^{-1} R_\eta(D_\eta) K_\eta(D_\eta)^{-1} \int K_\eta\left(D_{z,j}(\boldsymbol{\theta}), D_{y,k}\right) K_\eta\left(D_{z,i}(\boldsymbol{\theta}), D_{y,l}\right)^{\mathsf{T}} p\left(\boldsymbol{\theta}\right) \mathrm{d}\boldsymbol{\theta} \right)
\end{aligned}
\tag{5.14}
$$

Where $W_\eta(D_y)$ and $R_\eta(D_y)$ are presented in Eqs. (5.15) and (5.16).

$$
W_\eta(D_y) = \left(H_\eta(D_y)^{\mathsf{T}} K_\eta(D_y)^{-1} H_\eta(D_y)\right)^{-1} \tag{5.15}
$$

$$
R_\eta(D_y) = H_\eta(D_y) W_\eta(D_y) H_\eta(D_y)^{\mathsf{T}} \tag{5.16}
$$

In order to perform BCBC the integrals in Eqs. (5.12) and (5.14) must be solved. When particular forms of mean $m_\eta(\cdot)$ and covariance function $K_\eta(\cdot, \cdot)$ for the emulator are chosen, along with a specific prior distribution for $\boldsymbol{\theta}$, these integrals have closed form solutions. Appendix A.4 outlines these integrals in closed form when $\boldsymbol{\theta} \sim \mathcal{N}\left(m_\theta, V_\theta\right)$ and a constant mean and SE covariance functions are implemented in the emulator.

The approximation of $p\left(\boldsymbol{z} \,|\, \boldsymbol{y}, \boldsymbol{\phi}\right)$ as a Gaussian with the unconditional mean and covariance defined in Eqs. (5.11) and (5.13) can be used to form the marginal likelihood shown in Eq. (5.17). This is formed by integrating out the hyperparameter of the model discrepancy mean function $\boldsymbol{\beta}_\delta$ in Eqs. (5.18) and (5.19).

$$p\left(\boldsymbol{\phi}_\delta, \sigma_n^2, \rho \,|\, \boldsymbol{d}, \boldsymbol{\phi}_\eta\right) \propto |V|^{-\frac{1}{2}} |W_\delta|^{\frac{1}{2}}$$
$$\exp\left(-\frac{1}{2}\left(\boldsymbol{z} - H_\delta \hat{\boldsymbol{\beta}}_\delta - \rho\hat{\eta}(D_z)\right) V^{-1} \left(\boldsymbol{z} - H_\delta \hat{\boldsymbol{\beta}}_\delta - \rho\hat{\eta}(D_z)\right)\right) p\left(\boldsymbol{\phi}_\delta, \sigma_n^2, \rho\right) \quad (5.17)$$

$$\hat{\boldsymbol{\beta}}_\delta = W_\delta H_\delta(D_z)^\mathsf{T} V^{-1}(\boldsymbol{z} - \rho\hat{\eta}(D_z)) \quad (5.18)$$

$$W_\delta = \left(H_\delta(D_z)^\mathsf{T} V^{-1} H_\delta(D_z)\right)^{-1} \quad (5.19)$$

If $p\left(\boldsymbol{\phi}_\delta, \sigma_n^2, \rho\right)$ are Jeffrey's priors, then the estimates $\{\hat{\boldsymbol{\phi}}_\delta, \hat{\sigma_n^2}, \hat{\rho}\}$ from maximising Eq. (5.17) are MLE estimates — in practice, as discussed in Section 4.2, this is performed by minimising the NLML.

5.2.3 Calibration Parameter Inference

With the fixed set of hyperparameters $\hat{\boldsymbol{\phi}} = \{\hat{\boldsymbol{\phi}}_\eta, \hat{\boldsymbol{\phi}}_\delta, \hat{\sigma_n^2}, \hat{\rho}\}$ obtained from the two GP inference steps the joint Gaussian likelihood in Eqs. (5.5) and (5.6) can be formed. This joint Gaussian likelihood is conditioned on the fixed plug-in estimates $p(\boldsymbol{d} \,|\, \boldsymbol{\theta}, \boldsymbol{\phi} = \hat{\boldsymbol{\phi}})$. As discussed in Section 5.2 a full Bayesian analysis would require $\boldsymbol{\phi}$ to be integrated out so the posterior parameter distribution is dependent only on the data, however this integral is intractable. One solution is to take an empirical Bayes approach, whereby the hyperparameters are fixed at their MLE estimates $\hat{\boldsymbol{\phi}}$, rather than integrating them out numerically. This approach is taken here in order to keep the technique computationally efficient.

Calibration of the parameters $\boldsymbol{\theta}$ is performed using Bayesian inference where the joint posterior distribution is shown in Eq. (5.20).

$$p(\boldsymbol{\theta} \,|\, \boldsymbol{d}, \hat{\boldsymbol{\phi}}) \propto |\boldsymbol{V}_d(\boldsymbol{\theta})|^{-\frac{1}{2}} \exp\left(-\frac{1}{2}\left(\boldsymbol{d} - \boldsymbol{m}_d(\boldsymbol{\theta})\right) \boldsymbol{V}_d(\boldsymbol{\theta})^{-1} \left(\boldsymbol{d} - \boldsymbol{m}_d(\boldsymbol{\theta})\right)\right) p\left(\boldsymbol{\theta}\right) \quad (5.20)$$

As before, using non-informative priors $\boldsymbol{\beta} = \{\boldsymbol{\beta}_\eta^\mathsf{T}, \boldsymbol{\beta}_\delta^\mathsf{T}\}$ can be marginalised out of

Eq. (5.20) resulting in Eqs. (5.21) to (5.23) — for notational purposes the hyperparameter set becomes $\hat{\boldsymbol{\phi}} = \{\hat{\sigma}_\eta^2, \hat{\Omega}_x, \hat{\Omega}_t, \hat{\sigma}_\delta^2, \hat{\boldsymbol{\psi}}, \hat{\sigma_n^2}, \hat{\rho}\}$.

$$p(\boldsymbol{\theta} \,|\, \boldsymbol{d}, \hat{\boldsymbol{\phi}}) \propto |\boldsymbol{V}_d(\boldsymbol{\theta})|^{-\frac{1}{2}} |W(\boldsymbol{\theta})|^{\frac{1}{2}} \exp\left(-\frac{1}{2}\left(\boldsymbol{d} - H(\boldsymbol{\theta})\hat{\boldsymbol{\beta}}\right)\boldsymbol{V}_d(\boldsymbol{\theta})^{-1}\left(\boldsymbol{d} - H(\boldsymbol{\theta})\hat{\boldsymbol{\beta}}\right)\right) p\left(\boldsymbol{\theta}\right) \quad (5.21)$$

$$\hat{\boldsymbol{\beta}}(\boldsymbol{\theta}) = W(\boldsymbol{\theta}) H(\boldsymbol{\theta})^{\mathsf{T}} \boldsymbol{V}_d(\boldsymbol{\theta})^{-1} \boldsymbol{d} \quad (5.22)$$

$$W(\boldsymbol{\theta}) = \left(H(\boldsymbol{\theta})^{\mathsf{T}} \boldsymbol{V}_d(\boldsymbol{\theta})^{-1} H(\boldsymbol{\theta})\right)^{-1} \quad (5.23)$$

Equation (5.21) can be used to make inference about $\boldsymbol{\theta}$. Despite the construction of Eq. (5.21) full Bayesian analysis, which requires the evaluation of the marginal $\int p(\boldsymbol{d} \,|\, \boldsymbol{\theta}, \hat{\boldsymbol{\phi}}) p\left(\boldsymbol{\theta}\right) \mathrm{d}\boldsymbol{\theta}$, remains intractable due to the likelihood's dependence on $\boldsymbol{\theta}$ (shown in Eq. (5.21)). This means that numerical methods are utilised. Here two techniques are investigated, quadrature and MCMC sampling outlined in Sections 5.2.5 and 5.2.6.

5.2.4 Calibrated Predictive Posterior

The conditional distribution $p(\boldsymbol{z}_* \,|\, \boldsymbol{d}, \hat{\boldsymbol{\phi}}, \boldsymbol{\theta})$ for predicting n_* new observations $\boldsymbol{z}_*$ from new input locations $D_{z*} = \{\boldsymbol{x}_1^{z*}, \ldots, \boldsymbol{x}_{n_*}^{z*}\}$ is a GP (formed from standard Gaussian conditionals using Eqs. (5.5) and (5.6)). The posterior mean and covariance of the GP model are presented in Eqs. (5.24) and (5.25).

$$\mathbb{E}\left(\boldsymbol{z}_* \,|\, \boldsymbol{d}, \hat{\boldsymbol{\phi}}, \boldsymbol{\theta}\right) = H_*(D_{z*}(\boldsymbol{\theta}))\hat{\boldsymbol{\beta}} + V_*(D_{z*}(\boldsymbol{\theta}))^{\mathsf{T}} \boldsymbol{V}_d(\boldsymbol{\theta})^{-1}(\boldsymbol{d} - H(\boldsymbol{\theta})\hat{\boldsymbol{\beta}}) \quad (5.24)$$

$$\mathbb{V}\left(\boldsymbol{z}_* \,|\, \boldsymbol{d}, \hat{\boldsymbol{\phi}}, \boldsymbol{\theta}\right) = \rho^2 K_\eta(D_{z*}(\boldsymbol{\theta})) + K_\delta(D_{z*}) - V_*(D_{z*}(\boldsymbol{\theta}))^\mathsf{T} \boldsymbol{V}_d(\boldsymbol{\theta})^{-1} V_*(D_{z*}(\boldsymbol{\theta})) \\ \left(H_*(D_{z*}(\boldsymbol{\theta})) - H(\boldsymbol{\theta})^\mathsf{T} \boldsymbol{V}_d(\boldsymbol{\theta})^{-1} V_*(D_{z*}(\boldsymbol{\theta}))\right)^\mathsf{T} \\ W(\boldsymbol{\theta}) \left(H_*(D_{z*}(\boldsymbol{\theta})) - H(\boldsymbol{\theta})^\mathsf{T} \boldsymbol{V}_d(\boldsymbol{\theta})^{-1} V_*(D_{z*}(\boldsymbol{\theta}))\right) \quad (5.25)$$

Where the predictive design matrix and cross covariance terms are shown in Eqs. (5.26) and (5.27) respectively.

$$H_*(D_{z*}(\boldsymbol{\theta})) = \begin{bmatrix} \rho H_\eta(D_{z*}(\boldsymbol{\theta})) & H_\delta(D_{z*}) \end{bmatrix} \quad (5.26)$$

$$V_*(D_{z*}(\boldsymbol{\theta})) = \begin{bmatrix} \rho K_\eta(D_{z*}(\boldsymbol{\theta}), D_y) \\ \rho^2 K_\eta(D_{z*}(\boldsymbol{\theta}), D_z(\boldsymbol{\theta})) + K_\delta(D_{z*}, D_z) \end{bmatrix} \quad (5.27)$$

The predictive GP from Eqs. (5.24) and (5.25) is dependent on $\boldsymbol{\theta}$. To make calibrated predictions the unconditional predictive posterior $p(\boldsymbol{z}_* \,|\, \boldsymbol{d}, \hat{\boldsymbol{\phi}})$ is calculated by the marginalisation integral $\int p(\boldsymbol{z}_* \,|\, \boldsymbol{d}, \hat{\boldsymbol{\phi}}, \boldsymbol{\theta}) p(\boldsymbol{\theta} \,|\, \boldsymbol{d}, \hat{\boldsymbol{\phi}}) \mathrm{d}\boldsymbol{\theta}$. As a Gaussian distribution is fully specified by its first and second moments the integral is presented using the law of total expectation and covariance in Eqs. (5.28) and (5.29) (where $\mathbb{E}_\Theta(\cdot)$ and $\mathrm{cov}_\Theta(\cdot, \cdot)$ are the expectation and covariance with respect to the posterior distribution $p(\boldsymbol{\theta} \,|\, \boldsymbol{d}, \hat{\boldsymbol{\phi}})$).

$$\mathbb{E}\left(\boldsymbol{z}_* \,|\, \boldsymbol{d}, \hat{\boldsymbol{\phi}}\right) = \mathbb{E}_\Theta\left(\mathbb{E}\left(\boldsymbol{z}_* \,|\, \boldsymbol{d}, \hat{\boldsymbol{\phi}}, \boldsymbol{\theta}\right)\right) \quad (5.28)$$

$$\mathbb{V}\left(\boldsymbol{z}_* \,|\, \boldsymbol{d}, \hat{\boldsymbol{\phi}}\right) = \mathbb{E}_\Theta\left(\mathbb{V}\left(\boldsymbol{z}_* \,|\, \boldsymbol{d}, \hat{\boldsymbol{\phi}}, \boldsymbol{\theta}\right)\right) + \mathrm{cov}_\Theta\left(\mathbb{E}\left(\boldsymbol{z}_* \,|\, \boldsymbol{d}, \hat{\boldsymbol{\phi}}, \boldsymbol{\theta}\right), \mathbb{E}\left(\boldsymbol{z}'_* \,|\, \boldsymbol{d}, \hat{\boldsymbol{\phi}}, \boldsymbol{\theta}\right)\right) \quad (5.29)$$

The unconditional posterior mean is formed from the integral in Eq. (5.30).

$$\mathbb{E}\left(\boldsymbol{z}_* \,|\, \boldsymbol{d}, \hat{\boldsymbol{\phi}}\right) = \int H_*(D_{z*}(\boldsymbol{\theta}))\hat{\boldsymbol{\beta}} + V_*(D_{z*}(\boldsymbol{\theta}))^\mathsf{T}\boldsymbol{V}_d(\boldsymbol{\theta})^{-1}(\boldsymbol{d} - H(\boldsymbol{\theta})\hat{\boldsymbol{\beta}}) \\ p\left(\boldsymbol{d} \,|\, \boldsymbol{\theta}, \hat{\boldsymbol{\phi}}\right)p\left(\boldsymbol{\theta}\right)\mathrm{d}\boldsymbol{\theta} \quad (5.30)$$

The unconditional posterior covariance is formed from the integral in Eq. (5.31).

$$\begin{aligned}\mathbb{V}\left(\boldsymbol{z}_* \,|\, \boldsymbol{d}, \hat{\boldsymbol{\phi}}\right) = \int \Big[&\rho^2 K_\eta(D_{z*}(\boldsymbol{\theta})) + K_\delta(D_{z*}) - V_*(D_{z*}(\boldsymbol{\theta}))^\mathsf{T}\boldsymbol{V}_d(\boldsymbol{\theta})^{-1}V_*(D_{z*}(\boldsymbol{\theta})) \\ &\left(H_*(D_{z*}(\boldsymbol{\theta})) - H(\boldsymbol{\theta})^\mathsf{T}\boldsymbol{V}_d(\boldsymbol{\theta})^{-1}V_*(D_{z*}(\boldsymbol{\theta}))\right)^\mathsf{T} \\ &W(\boldsymbol{\theta})\left(H_*(D_{z*}(\boldsymbol{\theta})) - H(\boldsymbol{\theta})^\mathsf{T}\boldsymbol{V}_d(\boldsymbol{\theta})^{-1}V_*(D_{z*}(\boldsymbol{\theta}))\right) + \\ &\mathbb{E}\left(\boldsymbol{z}_* \,|\, \boldsymbol{d}, \hat{\boldsymbol{\phi}}, \boldsymbol{\theta}\right)\mathbb{E}\left(\boldsymbol{z}_* \,|\, \boldsymbol{d}, \hat{\boldsymbol{\phi}}, \boldsymbol{\theta}\right)^\mathsf{T}\Big]p\left(\boldsymbol{d} \,|\, \boldsymbol{\theta}, \hat{\boldsymbol{\phi}}\right)p\left(\boldsymbol{\theta}\right)\mathrm{d}\boldsymbol{\theta} + \\ &\mathbb{E}\left(\boldsymbol{z}_* \,|\, \boldsymbol{d}, \hat{\boldsymbol{\phi}}\right)\mathbb{E}\left(\boldsymbol{z}_* \,|\, \boldsymbol{d}, \hat{\boldsymbol{\phi}}\right)^\mathsf{T} \quad (5.31)\end{aligned}$$

Again due to the intractable nature of $p(\boldsymbol{\theta} \,|\, \boldsymbol{d}, \hat{\boldsymbol{\phi}})$ and the predictive GP posterior, the integrals in Eqs. (5.30) and (5.31) are solved numerically via quadrature or from performing Monte Carlo averaging from the MCMC sampled posterior distribution. Solving these integrals forms the unconditional posterior $p(\boldsymbol{z}_* \,|\, \boldsymbol{d}, \hat{\boldsymbol{\phi}})$ and therefore a calibrated prediction.

5.2.5 Gauss-Hermite Quadrature

There are several quadrature methods for approximating integrals with respect to various distributions. Gauss-Hermite quadrature is a particular form used for approximating integrals with respect to a Gaussian distribution [147], useful for the BCBC formulation outlined. The Gaussian quadrature integral can be approximated as,

$$\int_{-\infty}^{\infty} e^{-x^2} f(x)dx \approx \sum_{i=1}^{n} w_i f(x_i) \quad (5.32)$$

where $f(x)$ is the function to be integrated with respect to e^{-x^2} and w_i and x_i are a

set of n weights and nodes. The function e^{-x^2} is the weight function $w(x)$, which is captured in the w_i coefficients. The nodes x_i are the roots of special orthogonal polynomials called Hermite polynomials $H_p(x)$. In Gauss-Hermite quadrature the physicists' Hermite polynomial defined in Eq. (5.33) is used which has the weights shown in Eq. (5.34).

$$H_p(x) = p! \sum_{n=0}^{\lfloor \frac{p}{2} \rfloor} \frac{(-1)^n}{n!(p-2n)!} (2x)^{p-2n} \tag{5.33}$$

$$w_i = \frac{2^{p-1} p! \sqrt{\pi}}{p^2 (H_{p-1}(x_i))^2} \tag{5.34}$$

Where p is the degree of the polynomial and $\lfloor \cdot \rfloor$ is a floor function (i.e. the lowest integer is used). To determine these weights and nodes the Golub-Welsch algorithm can be used [148], presented in Appendix A.5.

To evaluate an integral with respect to a univariate Gaussian distribution with a mean μ and variance σ^2 the Gauss-Hermite quadrature is transformed into Eq. (5.35).

$$\mathbb{E}[f(X)] = \int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} f(x) dx \approx \sum_{i=1}^{n} \frac{w_i}{\sqrt{\pi}} f(\mu + \sqrt{2}\sigma x_i) \tag{5.35}$$

For the multivariate case where x is D dimensional the integral can be split into D nested Gauss-Hermite integrals forming Eq. (5.36).

$$\int \mathcal{N}(\boldsymbol{x} \mid \boldsymbol{\mu}, \Sigma) f(\boldsymbol{x}) \mathrm{d}\boldsymbol{x} \approx \sum_{i_1, i_2, \cdots, i_D} \pi^{-\frac{D}{2}} w_{i_1} w_{i_2} \cdots w_{i_D} f(\boldsymbol{\mu} + \sqrt{2} L(x_{i_1}, x_{i_2}, \cdots, x_{i_D})) \tag{5.36}$$

Where L is the lower matrix of the Cholesky decomposition, where $\Sigma = LL^{\mathsf{T}}$.

A problem with this method is that as D increases the number of points required grows exponentially, i.e. the *curse of dimensionality*. This means that the method is most applicable in low dimensional scenarios.

5.2.6 Markov Chain Monte Carlo

MCMC based techniques seek to obtain valid samples from intractable integrals or posterior distributions. This is performed by generating Markov chains that have the same stationary distribution as that of the posterior density. These methods rely on Markov chains, which satisfy the Markov property — a set of random variables $\boldsymbol{X}$, for which the state X_t has a conditional distribution given all previous states $X_1, \ldots, X_{t-1}$, that depends only on the previous state X_{t-1}. The aim is to generate stationary Markov chains whose equilibrium probability distributions are particular target distributions, and therefore obtain samples from these target distributions. The Metropolis-Hastings algorithm offers a method for constructing these Markov chains and is outlined in Algorithm 3 [149, 150].

The algorithm, for estimating intractable posterior distributions, requires the specification of the proposal distribution $q\left(\boldsymbol{\theta}^* \,|\, \boldsymbol{\theta}^{i-1}\right)$ and the known unnormalised density (i.e. $p\left(\boldsymbol{d} \,|\, \boldsymbol{\theta}\right)p\left(\boldsymbol{\theta}\right)$) in the acceptance kernel $\alpha(\boldsymbol{\theta}^*|\boldsymbol{\theta}^{i-1})$. When the proposal distribution is Gaussian distributed, i.e. $\mathcal{N}\left(\boldsymbol{\theta}^{i-1}, V\right)$, the Metropolis-Hastings random walk algorithm in Algorithm 1 is formed. The symmetric proposal means that the acceptance kernel becomes the ratio of the unnormalised density for the candidate over the previous sample in the Markov chain.

The Metropolis-Hastings algorithm samples from the proposal rather than the posterior distribution directly. However a property of the technique is that if the Markov chains are run long enough they will converge to sampling the target posterior distribution. This initial period before convergence is known as burn-in — where the Markov chain is heavily influenced by the initial state— and samples up to this point are discarded. Another method for assessing whether the Markov chain has sampled the posterior distribution in an adequate manner requires evaluating the acceptance ratio — the percentage of accepted samples. The optimal acceptance ratio depends on the geometry of the target distribution. For Gaussian proposals the optimal asymptotic acceptance rate for a D-dimensional target distribution is 0.234 [151]. The proposal distribution should be tuned in order to approach this limit and provide a good level of mixing in the Markov chains. Lastly the autocorrelation of the Markov chains should be interrogated, as only the previous point should be correlated due to the Markov property. The $\hat{R}$ statistic offers another diagnostic where the variances within- and between-multiple Markov chains are assessed, where a large difference in these variances indicate non-convergence [149].

Algorithm 3 Metropolis-Hastings

Set a proposal $q\left(\boldsymbol{\theta}^* \mid \boldsymbol{\theta}^{i-1}\right)$
Set $\boldsymbol{\theta}^0$ ▷ Set the initial state in the Markov Chain

for $i = 1 : N$ **do**
  $\boldsymbol{\theta}^* = q\left(\boldsymbol{\theta}^* \mid \boldsymbol{\theta}^{i-1}\right)$ ▷ Propose a new candidate sample
  $\alpha(\boldsymbol{\theta}^* | \boldsymbol{\theta}^{i-1}) = \min\left\{1, \frac{q\left(\boldsymbol{\theta}^{i-1} \mid \boldsymbol{\theta}^*\right) p(\boldsymbol{d} \mid \boldsymbol{\theta}^*) p(\boldsymbol{\theta}^*)}{q\left(\boldsymbol{\theta}^* \mid \boldsymbol{\theta}^{i-1}\right) p\left(\boldsymbol{d} \mid \boldsymbol{\theta}^{i-1}\right) p\left(\boldsymbol{\theta}^{i-1}\right)}\right\}$ ▷ Acceptance probability
  $u^i \sim \mathcal{U}(0, 1)$
  if $u^i \leq \alpha(\boldsymbol{\theta}^* | \boldsymbol{\theta}^{i-1})$ **then**
    $\boldsymbol{\theta}^i = \boldsymbol{\theta}^*$ ▷ Accept the sample
  else
    $\boldsymbol{\theta}^i = \boldsymbol{\theta}^{i-1}$ ▷ Reject the sample
  end if
end for

The standard Metropolis-Hastings algorithm has no method for incorporating information about the posterior from the accepted values in the chain. An adaptive Metropolis algorithm provides a method for updating the proposal based on previous accepted samples [150, 152]. The means that the process is no longer Markovian, but still ergodic, as states in the chain do not solely depend on the previous state. The algorithm is presented in Algorithm 4.

The algorithm requires determining an update rate k which typically will provide higher acceptance rates and better mixing when it is low. The term $\varepsilon \mathbb{I}_D$ is sometimes incorporated to make the covariance positive definite, although ε can often be set to 0. At the first update step the covariance is calculated in the standard form, however this becomes inefficient for other steps as the number of states increases. Instead the covariance is calculated based on a Bayesian update of the Gaussian proposal distribution, as presented in Eqs. (5.37) and (5.38).

$$V_i = \frac{i-2}{i-1} V_{i-1} + \frac{2.38^2}{D(i-1)} \left((i-1) \bar{\boldsymbol{\theta}}^{i-2} \bar{\boldsymbol{\theta}}^{i-2\mathsf{T}} - i \bar{\boldsymbol{\theta}}^{i-1} \bar{\boldsymbol{\theta}}^{i-1\mathsf{T}} + \boldsymbol{\theta}^{i-1} \boldsymbol{\theta}^{i-1\mathsf{T}} + \varepsilon \mathbb{I}_D \right) \tag{5.37}$$

$$\bar{\boldsymbol{\theta}}^i = \frac{i-1}{i} \bar{\boldsymbol{\theta}}^{i-1} + \frac{1}{i} \boldsymbol{\theta}^{i-1} \tag{5.38}$$

Algorithm 4 Adaptive Metropolis

Set the proposal $q\left(\boldsymbol{\theta}^* \,|\, \boldsymbol{\theta}^{i-1}\right) = \mathcal{N}\left(\boldsymbol{\theta}^{i-1}, V_0\right)$
$R =$chol(V_0) ▷ Cholesky decomposition of V_0
Set $\boldsymbol{\theta}^0$ ▷ Set the initial state in the Markov Chain

for $i = 1 : N$ **do**
  $\boldsymbol{\theta}^* = \boldsymbol{\theta}^{i-1} + R\varepsilon$ where $\varepsilon \sim \mathcal{N}(0, 1)$ ▷ Take a random walk
  $r = \frac{p(\boldsymbol{z}\,|\,\boldsymbol{\theta}^*)p(\boldsymbol{\theta}^*)}{p\left(\boldsymbol{z}\,|\,\boldsymbol{\theta}^{i-1}\right)p\left(\boldsymbol{\theta}^{i-1}\right)}$ ▷ Compute the ratio
  $u^i \sim \mathcal{U}(0, 1)$
  if $u^i \leq \min(1, r)$ **then**
    $\boldsymbol{\theta}^i = \boldsymbol{\theta}^*$ ▷ Accept the sample
  else
    $\boldsymbol{\theta}^i = \boldsymbol{\theta}^{i-1}$ ▷ Reject the sample
  end if
  if $\text{mod}(i, k) = 0$ **then** ▷ Update step
    $V_i = \frac{2.38^2}{D}\text{cov}\left(\boldsymbol{\theta}^0, \ldots, \boldsymbol{\theta}^{i-1}\right) + \varepsilon\mathbb{I}_D$ ▷ Update proposal variance
    $R =$chol(V_i) ▷ Cholesky decomposition of V_i
  else
    $V_i = V_{i-1}$
  end if
end for

5.2.7 Numerical Example

To demonstrate the effectiveness of BCBC a numerical example is presented. Here a simulator predicts the natural frequency ω_n under varying tensions T and mass M, for a mass, tensioned wire system. The problem seeks to calibrate mass given the simulator models a centrally positioned mass located between two boundaries 1m apart (i.e. $l = 1$m) defined in Eq. (5.39).

$$\eta(x, \theta) = \omega_n(T, M) = \frac{1}{\pi}\sqrt{\frac{T}{Ml}} \tag{5.39}$$

The observations are collected from a mass, tensioned wire system where $a = 0.2$ and $b = 1 - 0.2$, i.e. the mass is offset. This demonstrates a level of missing physics within the process. Observations are therefore obtained from Eq. (5.40), where $e \sim \mathcal{N}(0, \sigma_n^2)$.

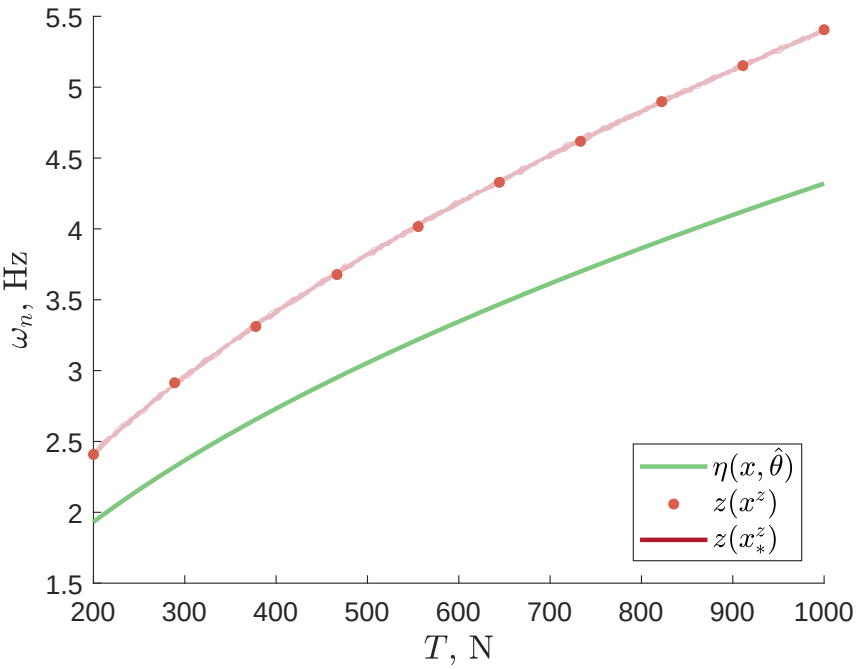


Figure 5.1: Difference between simulator and observations in the mass, tension wire system numerical study when the simulator with the 'true' mass is implemented.

$$z(x) = \frac{1}{2\pi}\sqrt{\frac{T(a+b)}{M(ab)}} + e \tag{5.40}$$

Figure 5.1 demonstrates the model discrepancy between the simulator and observations. In this example the noise variance was $\sigma_n^2 = 0.01^2$ and the inputs $\boldsymbol{x}^z = \{200, 288.9, \ldots, 1000\}$. BCBC was used to infer the parameter and predictive natural frequency distributions based on the observations at $\boldsymbol{x}^z$ using both Gauss-Hermite quadrature and adaptive Metropolis MCMC methodologies.

The emulator and model discrepancy hyperparameters were inferred and fixed before the two inference schemes, as stated in Section 5.2.3. The emulator was constructed from an SE covariance and constant mean functions with a nugget, $\nu = 1 \times 10^{-8}$. The model discrepancy GP was modelled with a Matérn covariance (where $p = 2$) and constant mean function. Figure 5.2 presents the inferred model discrepancy. These predictions are only possible when the 'true' model discrepancy is known, as in this numerical example, but would not be possible to visualise in most applications. The regression parameter ρ was 1.39 indicating the simulator was weighted more than the model discrepancy GP.

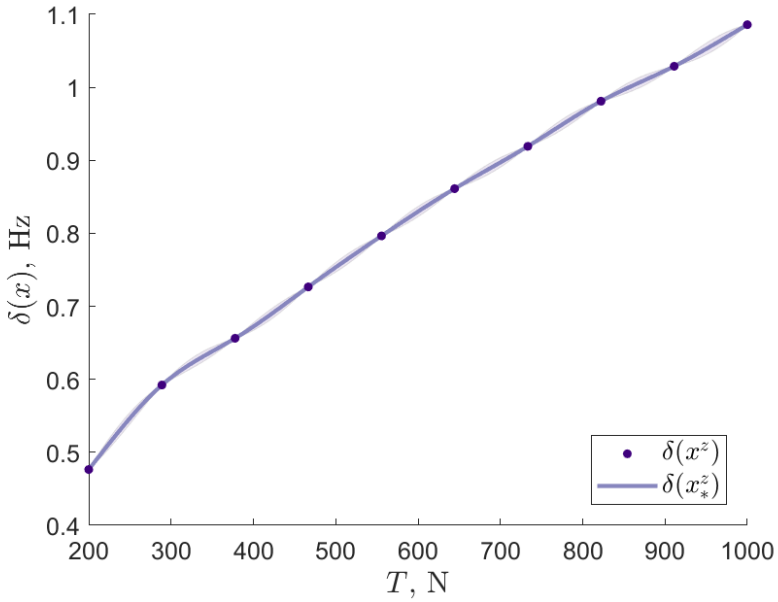


Figure 5.2: Inferred model discrepancy using BCBC for a mass, tensioned wire system. The shaded region indicates $\pm 3\sigma$.

For both techniques a prior for the mass was set as $\mathcal{N}(6, 1)$kg. Gauss-Hermite quadrature used 20 nodes and weights. The adaptive Metropolis algorithm was implemented with an update step size of 100. The burn in period was 1000 samples after which 10,000 posterior samples were obtained. The Markov chain was checked for ergodicity. The inferred parameter distributions from both approaches are presented in Fig. 5.3. Here it can be seen that the posterior distributions estimated by both the Gauss-Hermite quadrature[2] and adaptive Metropolis algorithms produce qualitatively similar distributions. The 'true' value is well within the modal mass of the two distributions, with the difference between the two statistical modes and the 'true' value being 0.07kg and 0.56kg for the Gauss-Hermite quadrature and adaptive Metropolis algorithms respectively. This shows that the methodology correctly identified the parameter in this case and that the Gauss-Hermite quadrature method performs better for this example.

Finally the predictions of natural frequency using the two inference approaches are shown in Fig. 5.4. 20 sets of test data were obtained for 100 equally spaced inputs, $x_*^z \in [200, 1000]$. The NMSE of the predictions at x_*^z from both approaches were 0.22,

[2]Due to implementation issues the Gauss-Hermite quadrature method produces a posterior distribution with an area less than one. To make this a valid PDF the posterior is scaled to sum to one.

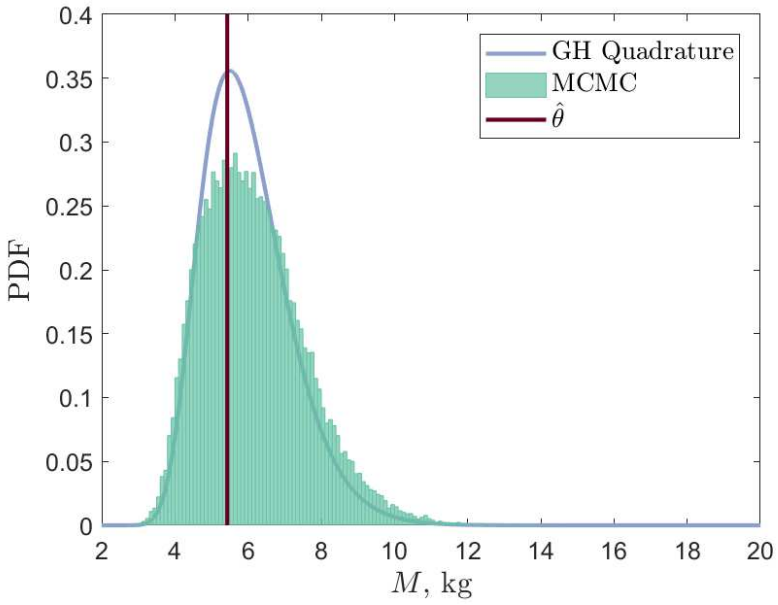


Figure 5.3: Posterior distributions using BCBC for a mass, tensioned wire system. Here GH stands for Gauss-Hermite and the adaptive Metropolis algorithm denoted MCMC.

demonstrating very good agreement in the mean prediction and that both inference schemes produce the same mean prediction.

The same numerical problem was repeated with $\sigma_n^2 = 0.1^2$ to assess the methods robustness to noise. The natural frequency predictions for BCBC using Gauss-Hermite quadrature are presented in Fig. 5.5a. It can be seen that the predictions have captured the increased noise, reflected in a NMSE of 1.59. However the parameter posterior distribution, displayed in Fig. 5.5b, is slightly further from the true value with a modal value of 5.89kg; although well within the probability mass. The simulator was again weighted highly as $\rho = 1.31$, similar to the inferred value when $\sigma_n^2 = 0.01^2$. These results indicate that non-identifiability issues become more pronounced in high noise scenarios. A potential solution would be to use multiple repeats for each observation in training, however this may not be practical for a forward model-driven SHM scenario.

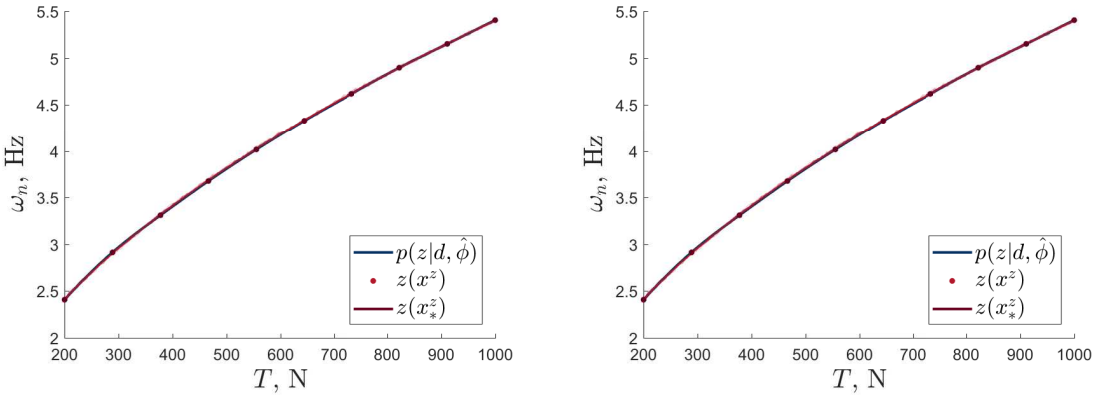


Figure 5.4: Predictions of natural frequency using BCBC for a mass, tensioned wire system. Panel (a) are the predictions using Gauss-Hermite quadrature and (b) using adaptive Metropolis MCMC. The shaded regions indicate $\pm 3\sigma$.

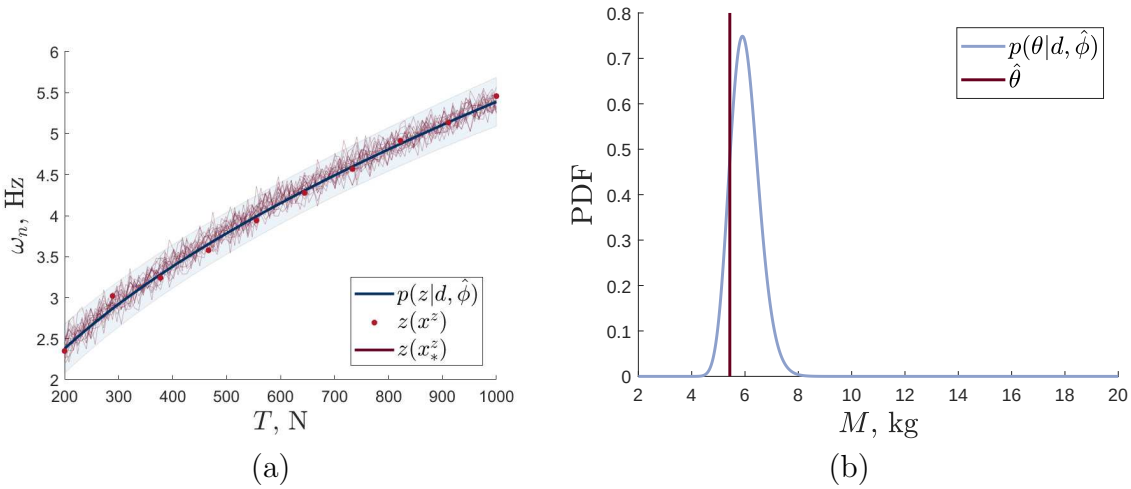


Figure 5.5: Predictions of natural frequency using BCBC for a mass, tensioned wire system where $\sigma_n^2 = 0.1^2$. Panel (a) are the predictions of natural frequency and (b) the posterior parameter distribution using Gauss-Hermite quadrature. The shaded region in panel (a) indicates $\pm 3\sigma$.

5.3 Representative Three Storey Building Case Study

BCBC and Bayesian calibration (without bias correction) were performed on a representative three storey building as an experimental case study. The aim was to indicate the improved accuracy of BCBC for forward model-driven SHM over conventional Bayesian calibration. Modal testing of the structure, presented in Fig. 5.6, was performed for nine damage extents — crack lengths of $\boldsymbol{x}_*^z = \{0, 2.5, \ldots, 20\}$mm in the front right beam in Fig. 5.6 — and the first three bending natural frequencies obtained. The structure was excited with broadband white noise via an electrodynamic shaker and the acceleration response measured at each of the three floors. Five repeats were obtained for each damage scenario. The third natural frequency was the most sensitive to damage and therefore used as the damage feature in this analysis. The experimental training data were five repeats when $\boldsymbol{x} = \{0, 5, 20\}$mm — chosen to indicate the methods effectiveness for identifying the functional form from a small number of observations. The validation data set included all five repeats for the nine damage extents.

The simulator was a modal FEA model where the saw cut was modelled geometrically, i.e. the geometry of the saw cut was included in that of the beam. The elastic modulus E was included in the calibration process. This meant that simulator evaluations for training the emulator were obtained at $\boldsymbol{x} = \boldsymbol{x}_*^z$ and $\boldsymbol{t} = \{65, 66, \ldots, 71\}$GPa due to a prior elastic modulus of $E \sim \mathcal{N}(68, 0.1)$GPa.

Figure 5.6: Experimental setup for the representative three storey building structure.

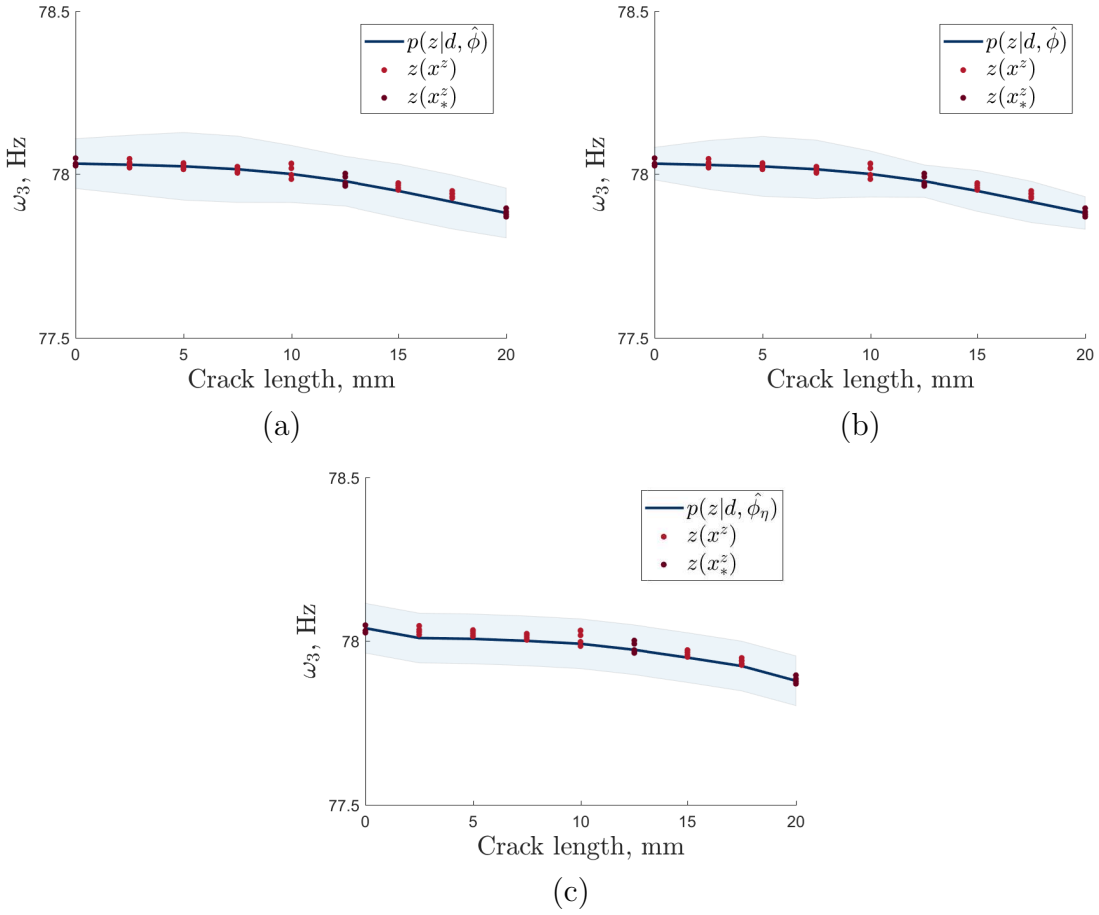


Figure 5.7: Predictions of natural frequency using BCBC and Bayesian calibration for a three storey building structure. Panel (a) and (b) are BCBC predictions using Gauss-Hermite quadrature and adaptive Metropolis MCMC respectively. Panel (c) demonstrates Bayesian calibration using adaptive Metropolis MCMC. The shaded regions indicate $\pm 3\sigma$.

BCBC was performed using both the Gauss-Hermite quadrature (20 nodes and weights) and adaptive Metropolis MCMC inference methods. These results were compared to Bayesian calibration using a Gaussian likelihood with an unknown noise variance. The noise variance had a Gaussian prior, $\sigma_n^2 \sim \mathcal{N}(0.0044, 0.0001)$ where the mean was estimated from the variance of training observations $\mathbb{V}(z(\boldsymbol{x}))$. A GP emulator, fitted to the same simulator training data, was used to assess the likelihood — where the likelihood covariance was the summation of the emulator covariance and a diagonal matrix of σ_n^2, i.e. $\mathbb{I}\sigma_n^2$. Inference was performed using adaptive Metropolis MCMC for the Bayesian calibration approach.

For both the Bayesian calibration and BCBC approaches the adaptive Metropolis MCMC parameters were $50,000$ posterior samples after a 1000 sample burn in and an

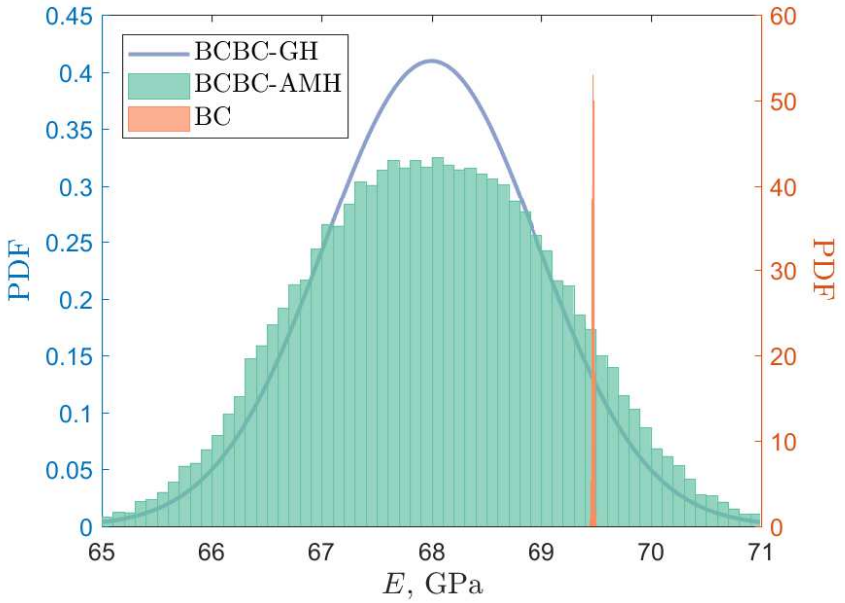


Figure 5.8: Posterior distributions for a three storey building structure using BCBC via Gauss-Hermite quadrature (BCBC-GH), adaptive Metropolis MCMC (BCBC-MCMC), and Bayesian calibration (BC) methods.

update step size of 100. The initial proposal variance for BCBC was 0.1 for the elastic modulus. On the other hand, the proposal covariance for Bayesian calibration had zero cross-covariance terms with a proposal variance of 0.02 for the elastic modulus and 0.01 for the noise variance. All these approaches defined an emulator with constant mean and SE covariance functions with a nugget $\nu = 1 \times 10^{-8}$. The BCBC methods were implemented with a model discrepancy prior defined by constant mean and Matérn (where $p = 2$) covariance functions.

The predictive distributions of the third natural frequency for all three approaches are displayed in Fig. 5.7. Here it can be seen that all three approaches have captured the trend of natural frequency with increased saw cut size, with the validation data lying within three standard deviations. The NMSE of the mean predictions for BCBC were both 8.07 compared to 12.34 for Bayesian calibration.

The inferred posterior parameter distributions are shown in Fig. 5.8. It can be seen that both the Gauss-Hermite quadrature and adaptive Metropolis MCMC methods produce similar posterior distributions. The variance of these distributions is large compared to the prior, and larger than the inferred posterior distribution from the Bayesian calibration approach. This difference between the BCBC and Bayesian

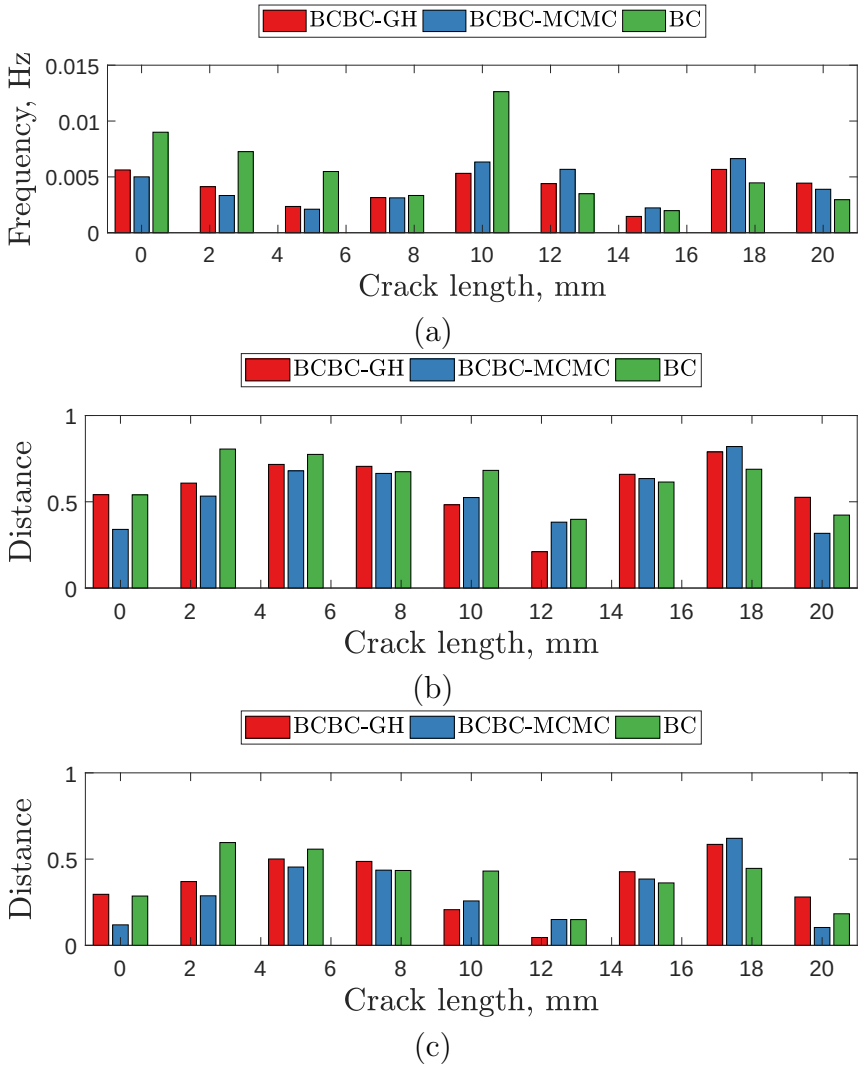


Figure 5.9: Validation metrics for natural frequency predictions from the three storey building case study. Panel (a), (b) and (c) demonstrate the area metric, total variation and Hellinger distances when compared to Gaussian representations of the observation data. Each panel demonstrates the distances for BCBC using Gauss-Hermite quadrature (BCBC-GH), adaptive Metropolis MCMC (BCBC-MCMC), and Bayesian calibration (BC) methods.

calibration posterior distributions is likely due to the omission of model discrepancy uncertainty in the Bayesian calibration formula, producing overconfident results. The regression parameter ρ was inferred as 0.08 from the BCBC approach, indicating model form errors due to a low weighting. This understanding should lead to model improvement where ρ should subsequently increase, reflecting a simulator that better captures the physics. It can be seen in Fig. 5.7c that these model form errors exist, noted by the functional difference between the 0 and 2.5mm damage extents, leading to under-estimation of the mean for other damage extents.

Method	0.0mm	2.5mm	5.0mm	7.5mm	10.0mm
BCBC-GH	0	0	0	0	0
BCBC-MCMC	0	0	0	0	0
BC	0	1	1	1	1

Method	12.5mm	15.0mm	17.5mm	20.0mm
BCBC-GH	0	1	1	0
BCBC-MCMC	0	1	1	0
BC	0	0	1	0

Table 5.1: KS-test results for the three storey case study where $\alpha = 0.05$. The hypothesis tests were applied to the BCBC using Gauss-Hermite quadrature (BCBC-GH), adaptive Metropolis MCMC (BCBC-MCMC), and Bayesian calibration (BC) predictions.

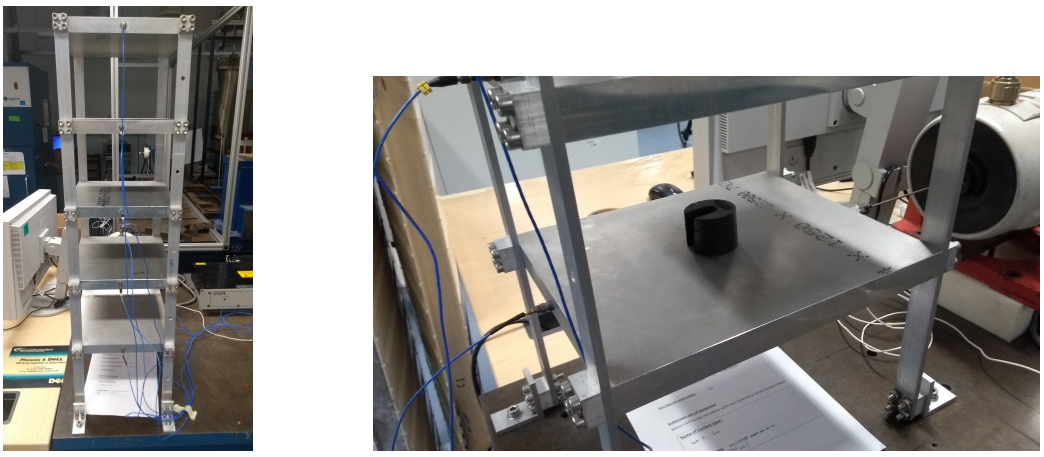
Hypothesis testing using the KS-test (and a significance level $\alpha = 0.05$), shown in Table 5.1 revealed that all output predictive distributions for BCBC, using both the Gauss-Hermite quadrature and adaptive Metropolis MCMC, produced the same hypothesis test results. This demonstrates the similarity in inference approximations. The null hypothesis was rejected for the 15.0 and 17.5mm damage extents only, stating a good predictive performance. The rejection of the null hypothesis for these predictions is likely due to an offset in mean prediction, as shown in Fig. 5.7a and Fig. 5.7b. In contrast, five damage state predictions using Bayesian calibration had significant statistical differences leading to a rejection of the null hypothesis. This indicates the issues due to model form errors, which are visually present in Fig. 5.7c.

The area metric, total variation and Hellinger distances were quantified and displayed in Fig. 5.9. The area metric shows the large distances for the Bayesian calibration predictions in the first five damage extents, compared with the BCBC approaches. The total variation and Hellinger distances indicate quite even predictive quality between all three methods, with BCBC using adaptive Metropolis MCMC slightly outperforming the other two approaches. As a result it can be determined that although improvements are evident from both BCBC methods over Bayesian calibration alone, they are not consistently better across all individual damage states.

5.4 Representative Five Storey Building Case Study

A second case study where a representative five story building structure was subject to pseudo-damage, via masses attached to the first floor, was calibrated using BCBC. This demonstrates that BCBC is applicable for multiple parameter calibration, typical in forward model-driven applications. The experimental structure made from aluminium 6082, displayed in Fig. 5.10, was subject to modal testing for six different masses, $m = \{0, 0.1, \ldots, 0.5\}$kg, with the first five bending modes extracted. Gaussian white noise, with a bandwidth of 409.6Hz and a chosen frequency resolution of 0.05Hz was implemented in exciting the structure via an electrodynamic shaker, with accelerators placed at each of the five floors. 40 averages were obtained for each measurement with ten repeats at each damage extent.

The observational training data included three mass scenarios $\boldsymbol{x}^z = \{0, 0.3, 0.5\}$kg, where only the first two (out of the ten repeats) were used to form the training set $z(\boldsymbol{x}^z)$. The remaining observations were incorporated in a validation set $z(\boldsymbol{x}^z_*)$. This reduced training data set demonstrates the ability of BCBC to capture the functional behaviour with a small subset of damage state data. The simulator $\eta(\boldsymbol{x}, \boldsymbol{t})$, a modal FEA model, modelled the five bending natural frequencies under the six damage extents $\boldsymbol{x} = \{0, 0.1, \ldots, 0.5\}$kg — displayed in Fig. 5.11. The FEA model did not model the complete bolted joint but simplified the joints by defining the each beam as fixed to each floors; adding an element of known model discrepancy.



(a) (b)

Figure 5.10: Representative five storey building structure. Panel (a) show the test setup and panel (b) presents an example of the pseudo-damage, added masses, applied to the first floor.

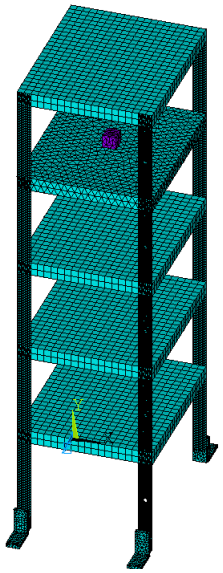


Figure 5.11: FEA model of the representative five storey building structure.

A fifty point, three dimensional GMLHD was constructed such that an emulator could be established over the parameter $\boldsymbol{t}$ space; where the parameters were elastic modulus E, Poisson's ratio ν and density ρ. Prior beliefs for each of these parameters were $E \sim \mathcal{N}(75, 1)$, $\nu \sim \mathcal{N}(0.32, 0.0001)$ and $\rho \sim \mathcal{N}(2800, 1000)$, reflecting typical properties of aluminium 6082.

Each of the five natural frequencies were calibrated using independent BCBC models with the same input properties. The emulators were constructed from constant mean and SE covariance functions with a nugget, $\nu = 1 \times 10^{-8}$. The model discrepancy GP priors were constant mean and Matérn covariance (where $p = 2$) functions. Inference was performed via adaptive Metropolis MCMC where 10000 posterior samples were obtained after a 1000 sample burn in and an update step at ever 100 accepted samples.

The five natural frequency predictions are displayed in Fig. 5.12 where the NMSEs were 176.73, 0.07, 0.01, 0.02, 0.11 and the log posterior likelihoods 210.6, 201.4, 273.2, 222.2, 260.0 respectively. These results show that the mean trend was captured for the second to fourth natural frequencies and indicate poor mean predictions for the first natural frequency. This is likely due to relatively low signal information being contained within the observation data and that no information about the mean for the 0.1 and 0.2kg damage states was contained within the training data. In contrast the log posterior likelihoods state that the validation data for all five natural frequencies could plausibly have been generated from the BCBC predictive

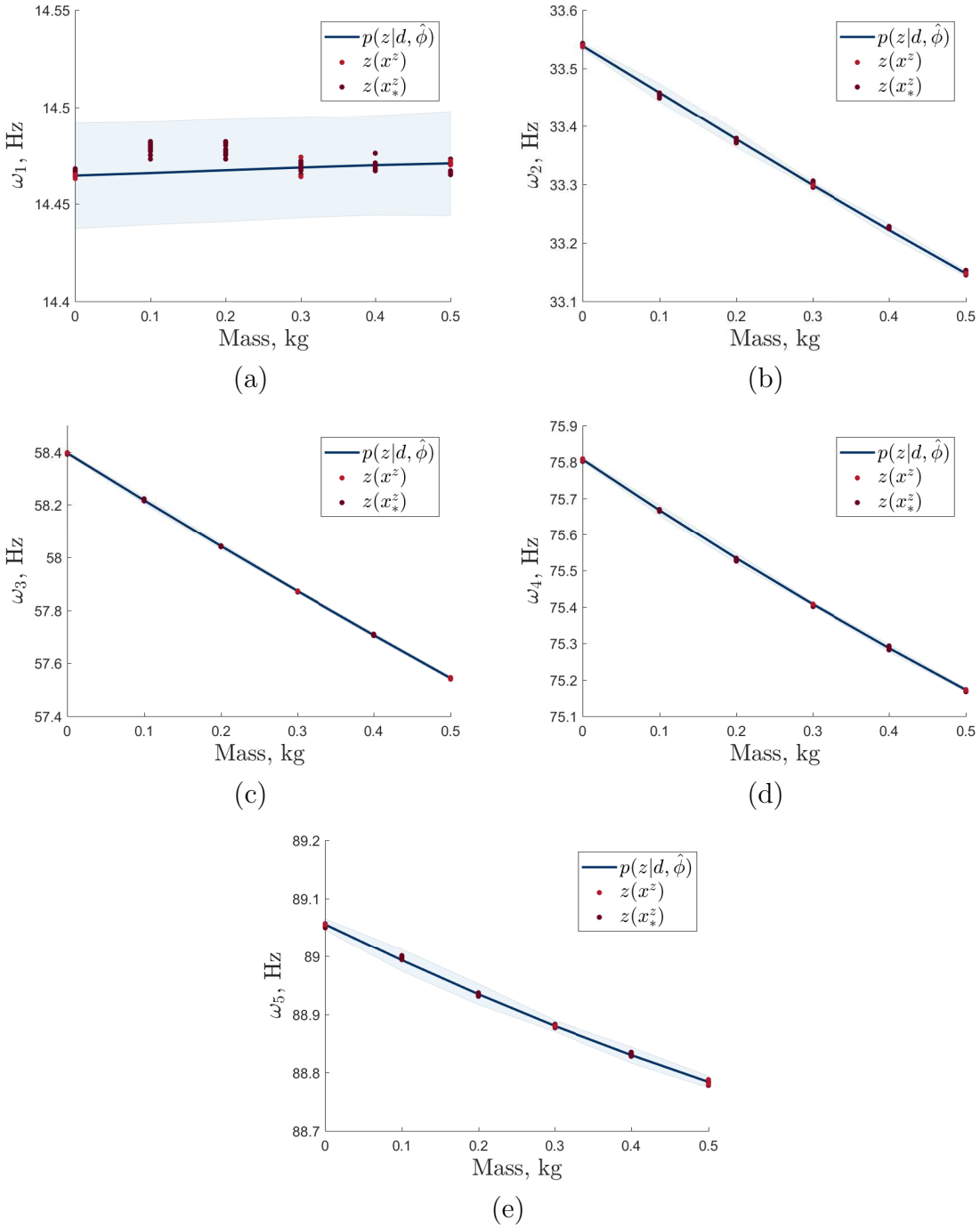


Figure 5.12: Predictions of natural frequency using BCBC for a five storey building structure. Panel (a), (b), (c), (d) and (e) are the first, second, third, fourth and fifth natural frequencies respectively. The shaded regions indicate $\pm 3\sigma$.

distributions.

In addition, Fig. 5.13 presents the inferred posterior parameter distributions. Qualitatively these distributions for elastic modulus, Poisson's ratio and density are very similar for the five natural frequencies, informing that the BCBC inferences are consistent. Furthermore, the regression parameter for each of the five natural frequency predictions were 0.30, 0.75, 1.03, 1.20, 1.64 which can be interpreted as stating the simulator performance is poorest for the first natural frequency with the second natural frequency also indicating problems. The third to fifth natural frequencies are adequately capture by the simulator and therefore have been weighted more highly. This information should result in improved model development targeting the first and second natural frequencies.

Hypothesis testing using both KS- and MMD two sample tests were performed to assess whether the observations could plausibly have been drawn from the predicted distributions, with a significance level $\alpha = 0.05$. 100 repeats of the MMD hypothesis test were performed (due to the predictive distributions being sampled ten times) for this particular test. Both the KS- and MMD hypothesis tests fail to reject the null hypothesis for 50% and 50.3% of the predictions (where ≥ 0.5 is considered a rejection of the null hypothesis for the averaged MMD tests). This demonstrates a relatively good prediction quality. Moreover, the two types of hypothesis test present relatively consistent results further weighting the hypothesis tests conclusions.

Finally distance metrics were applied in order to quantify the differences between the observational and predictive distributions (where either empirical CDFs or KDEs were used). Figure 5.14 displays the area metric, total variation, Hellinger and averaged MMD distances (from 100 repeats). The area metric for all the predictions are low, $\leq 5 \times 10^{-3}$Hz, indicating a good prediction quality. Consistently across all the distance metrics, damage states 0.1 and 0.2kg for the first natural frequency show large distances. This is due to the mean offset in the predicted distributions. Hellinger, total variation and MMD distances show similar distance patterns between natural frequencies and the damage states. These distances inform that the 0.4kg state for the second natural frequency has a large distance between the predicted and observed distributions. This can be seen in Fig. 5.12b with the small offset in the predictive mean.

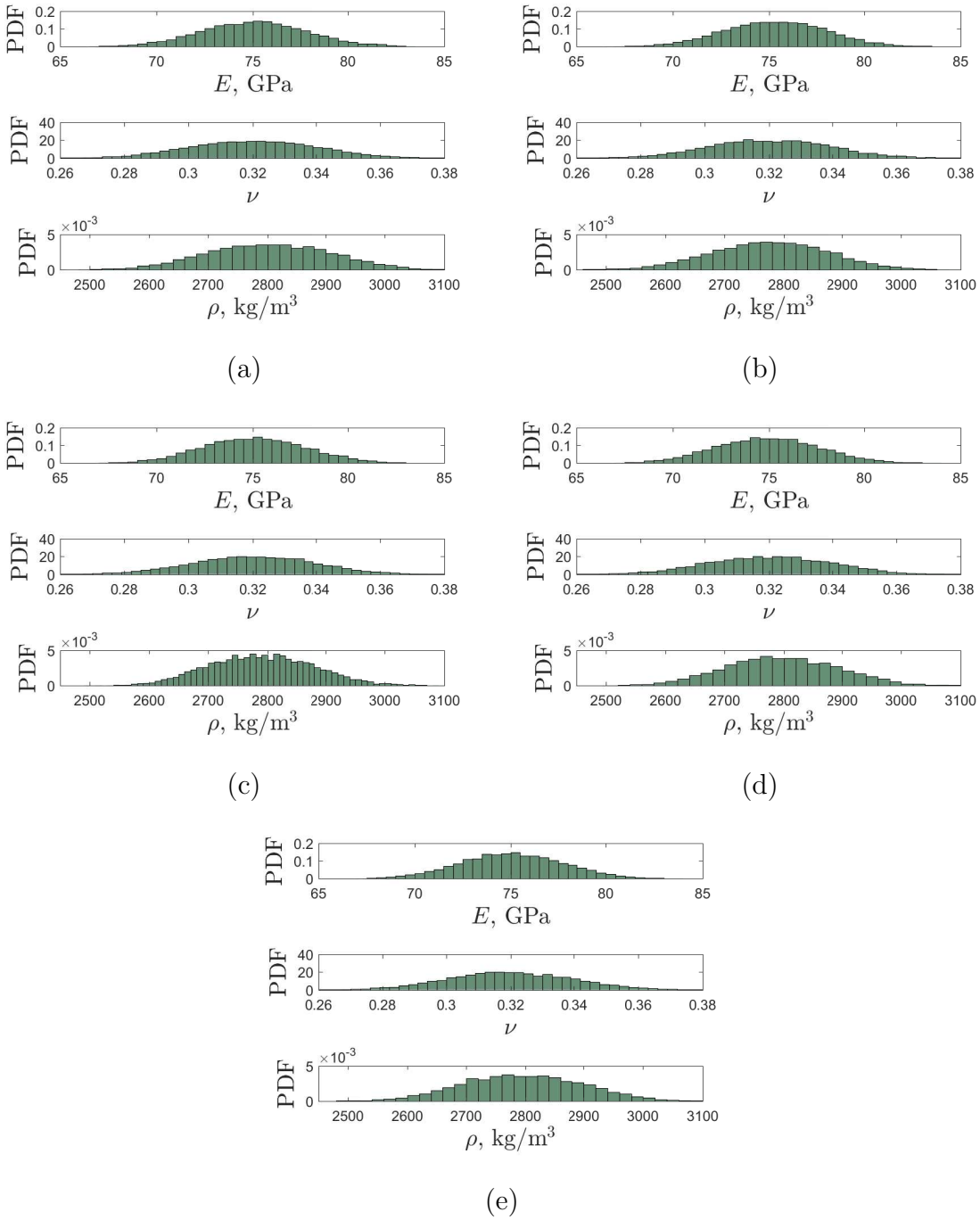


Figure 5.13: Posterior parameter distributions using BCBC for a five storey building structure. Panel (a), (b), (c), (d) and (e) are the first, second, third, fourth and fifth natural frequencies respectively.

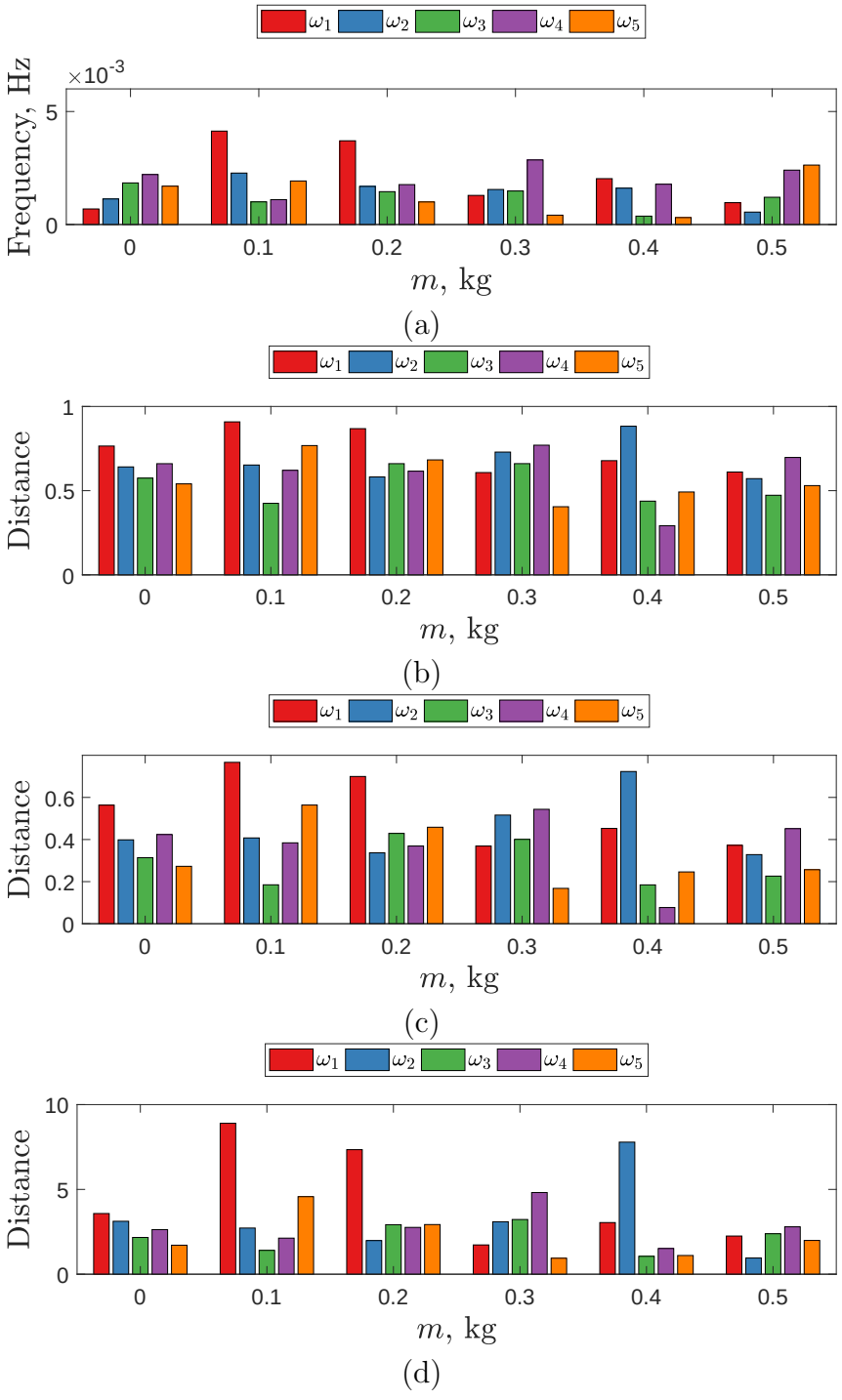


Figure 5.14: Statistical distances applied to the predictions from BCBC on the five storey building structure. Panel (a) is the area metric when compared to an empirical ten point observational CDF. Panel (b) and (c) are the total variation and Hellinger distances when compared to KDEs of the observational data. These three distance metrics have been calculated via numerical integration. Panel (d) is the averaged MMD distance over 100 repeats of ten samples from the predictive distribution.

Output	0.0kg	0.1kg	0.2kg	0.3kg	0.4kg	0.5kg
ω_1	0	1	1	0	0	0
ω_2	1	1	0	1	1	0
ω_3	1	0	1	1	0	1
ω_4	1	0	1	1	0	1
ω_5	0	1	0	0	0	0

Table 5.2: KS-test results for BCBC on the five storey building case study where $\alpha = 0.05$.

Output	0.0kg	0.1kg	0.2kg	0.3kg	0.4kg	0.5kg
ω_1	0.80	1.00	0.99	0.18	0.74	0.21
ω_2	0.64	0.56	0.34	0.73	0.96	0.05
ω_3	0.15	0.12	0.42	0.52	0.06	0.36
ω_4	0.47	0.29	0.55	0.86	0.12	0.54
ω_5	0.21	0.86	0.56	0.08	0.08	0.22

Table 5.3: MMD two sample test results for BCBC on the five storey building case study where $\alpha = 0.05$. Results are the average over 100 repeats of ten samples from the predictive distribution, using a bootstrap approach with ten shuffles and a squared exponential kernel where the hyperparameters are determined by a median heuristic.

5.5 Conclusion

BCBC has been demonstrated on several case studies in order to determine the methods applicability for forward model-driven SHM. The method has been shown to adequately correct model form errors in order to produce more statistically representative prediction than Bayesian calibration alone. Two inference methods have been demonstrated, Gauss-Hermite quadrature and adaptive Metropolis MCMC, which have been shown to be comparable in predictive quality.

Within a numerical case study increased noise was demonstrated to lead to misidentification of the parameter distribution, problematic if the inferred posterior distribution is used to inform other modelling steps within a forward model-driven strategy. Consequentially, the technique is most applicable in scenarios where either informative prior parameter information is known, or when the parameter distributions are not used to inform further modelling, i.e. only representative prediction are required. These findings agree with the conclusions of previous authors where the flexibility of the model discrepancy GP will lead to improved predictive quality but does not

guarantee correct parameter inference without strong prior information.

Further research should be conducted to see if there are improvements in predictive quality when multivariate GP priors are implemented (for both the emulator and model discrepancy). Subsequently, constraints on the model discrepancy GP should also be applied when known, unfortunately these constraints may be difficult to define in many applications.

A strength of the BCBC approach is that the regression parameter informs the level of model error, where below one states model form issues. This should be investigated within a simulator improvement strategy where the regression parameter identifies the parts of the simulator to target. On the other hand, a disadvantage of the approach is that the model discrepancy GP can only be visualised when the model discrepancy is known for the training data. This can be a limitation in knowing functionally how to improve the simulator.

Chapter 6

Bayesian History Matching

This chapter proposes an alternative approach to calibrating (or pre-calibrating) simulators whilst accounting for model discrepancy, namely BHM. The technique provides an alternative framework from standard Bayesian inference, is 'likelihood free', and can be seen as a special case of Approximate Bayesian Computation (ABC). BHM aims to reduce the parameter space by identifying and discarding simulator parameter combinations that were unlikely to have produced the observational outputs given the considered uncertainties.

The methodology is extended by considering techniques for incorporating sequential design of experiments within BHM using heuristics adapted from Bayesian optimisation. In addition, importance sampling based techniques are developed for inferring model discrepancy. This novel approach allows BHM to perform model discrepancy inference making it a competitive alternative to BCBC, whilst separating out parameter and model discrepancy inferences.

This chapter begins with a review of the BHM literature before outlining and extending BHM on numerical examples. A case study, using a five storey building structure is subsequently presented where the methodology for inferring model discrepancy via importance sampling is detailed and demonstrated. The results are validated before conclusions are provided.

6.1 Literature Review

History matching is a term that originates from the oil industry and describes methods that find parameters of simulators where the outputs closely match data from historical reservoir production. Many approaches within the literature using history matching as a term, as reviewed by Oliver and Chen [153], are similar to classical model updating techniques that are well-established within the SHM community [9]. Nonetheless, Craig et al. adapted the idea of history matching outlining a Bayesian methodology that searched for all, rather than a single parameter match [67] and defined this class of approaches as BHM. It is this form of history matching that is discussed within this chapter.

BHM begins by defining some form of criteria and metric for determining whether parameter combinations $\boldsymbol{\theta}$ are implausible, and not likely to have produced known observations $\boldsymbol{z}$. By discarding the implausible parameter space $\boldsymbol{\theta}_I \in \boldsymbol{\theta}$ the approach has a similar objective to calibration methods in that the remaining non-implausible space $\boldsymbol{\theta}_{nI} \in \boldsymbol{\theta}$ (parameters that provide acceptable matches given the criteria) are identified. The technique does not naturally provide a distribution over the non-implausible parameter set, however as described in Section 6.2.1, an approximation can be obtained. A key strength of the approach is, that by being 'likelihood free' inputs and outputs of the model can be included and excluded from each iteration without invalidating the analysis. This makes the technique a useful pre-calibration tool for a likelihood based calibration, such as MCMC based approaches, and can aid non-identifiability problems by informing more informed prior distributions.

BHM has been formulated and applied to a variety of applications from its origins in oil reservoir modelling [67] to understanding Galaxy formation [96, 154, 155], complex social models of HIV transfer in populations [71, 109] and climate science [156, 157]. In order to make the approach computationally efficient emulators are often implemented with common choices being GPs [71] and Bayes linear techniques [96, 155].

6.2 Methodology

BHM seeks to calibrate a statistical model of the form shown in Eq. (6.1).

$$z_j(\boldsymbol{x}) = \eta_j(\boldsymbol{x}, \boldsymbol{\theta}) + \delta_j + e_j \tag{6.1}$$

Where $z_j(\boldsymbol{x})$ is the jth observational output given inputs $\boldsymbol{x}$, $\eta_j(\boldsymbol{x}, \boldsymbol{\theta})$ is the jth simulator given $\boldsymbol{x}$ and parameters $\boldsymbol{\theta}$. The model discrepancy and observational uncertainty are δ and e respectively. The model assumes that the simulator, model discrepancy and observational uncertainty are independent and does not seek to define the model discrepancy's functional form.

In order to calibrate Eq. (6.1) the parameter space of the simulator is explored in iterations called waves. During a wave simulator outputs are assessed for parameter combinations and discarded based on a metric and threshold. This process would be prohibitively computationally expensive in most applications if simulator runs were required for each proposed parameter combination. To reduce this computational burden an emulator is implemented, with common techniques being GPs [71] and Bayes linear [96, 155] emulators — here for the reasons outlined in Section 4.1 a GP is utilised. The GP emulator is constructed as in Eq. (6.2).

$$\eta_j(\boldsymbol{x}, \boldsymbol{\theta}) \sim \mathcal{GP}_j\left(m(\boldsymbol{x}, \boldsymbol{\theta}), k((\boldsymbol{x}, \boldsymbol{\theta}), (\boldsymbol{x}', \boldsymbol{\theta}'))\right) \tag{6.2}$$

The predictive GP emulator mean $\mathbb{E}\left(\mathcal{GP}_j\left(\boldsymbol{x}, \boldsymbol{\theta}\right)\right)$ allows efficient assessment and exploration of the parameter space whilst also quantifying code uncertainty, $V_c(\boldsymbol{x}, \boldsymbol{\theta}) = \mathbb{V}\left(\mathcal{GP}_j\left(\boldsymbol{x}, \boldsymbol{\theta}\right)\right)$. The formulation stated in Eq. (6.2) assumes univariate GP emulators for each output, however multivariate GPs could be implemented (see Section 4.4.1 for details).

BHM employs a quantity that assesses the dissimilarities between observations and simulator outputs. A common metric is implausibility, which is the distance between observations and simulator outputs, weighted by the process's uncertainties, defined in Eq. (6.3).

$$I_j(\boldsymbol{x}, \boldsymbol{\theta}) = \frac{|z_j(\boldsymbol{x}) - \mathbb{E}\left(\mathcal{GP}_j\left(\boldsymbol{x}, \boldsymbol{\theta}\right)\right)|}{\left(V_{o,j} + V_{m,j} + V_{c,j}(\boldsymbol{x}, \boldsymbol{\theta})\right)^{1/2}} \tag{6.3}$$

Where, V_o, V_m and $V_c(\boldsymbol{x}, \boldsymbol{\theta})$ are the variances associated with the observational, model discrepancy and code uncertainties. By including code uncertainty $V_c(\boldsymbol{x}, \boldsymbol{\theta})$ into Eq. (6.3) parameter space is retained if the emulator variance is high for a

particular parameter combination, meaning that space is not discarded until the emulator is more certain that it accurately represents the simulator in that region. The observational uncertainty V_o can often be estimated from expert knowledge and from the observational data. Model discrepancy uncertainty V_m can be more challenging to define, but should be elicited from expert judgement; sensitivity analysis can be performed during a wave to understand changes in rejection rates. Observational and model discrepancy uncertainties can be dependant on both inputs $\boldsymbol{x}$ and outputs $z_j(\boldsymbol{x})$, i.e. $V_{o,j}(\boldsymbol{x})$ and $V_{m,j}(\boldsymbol{x})$, if input dependent hetroscedastic noise or model discrepancy are hypothesised.

The implausibility metric presented in Eq. (6.3) provides a quantity for every parameter combination, input and output, however a single value is required for each parameter combination in order to decide whether it should be removed. Several extensions of the implausibility metric that deal with multiple outputs and inputs can be considered. Firstly, a maximum implausibility can be formed, whereby the worst case for a given parameter combination is used, defined in Eq. (6.4).

$$I_{max}(\boldsymbol{\theta}) = \arg\max_j \left(\arg\max_{x_i} I_j(\boldsymbol{x}, \boldsymbol{\theta}) \right) \tag{6.4}$$

The other approach is to form a multivariate implausibility metric for either the inputs or outputs, Eqs. (6.5) and (6.6). This is equivalent to taking the Mahalanobis distance, standard practice in outlier analysis [158], which assesses the euclidean distance of the principle components. Again a maximum can be taken over either Eqs. (6.5) and (6.6) to collapse the metric to a single value for each parameter combination.

$$\begin{aligned} I_{multi}(\boldsymbol{\theta})_j = &\left(z_j(\boldsymbol{x}) - \mathbb{E}\left(\mathcal{GP}_j\left(\boldsymbol{x}, \boldsymbol{\theta}\right)\right)\right)^{\mathsf{T}} \\ &\left(V_{o,j} + V_{m,j} + V_{c,j}(\boldsymbol{x}, \boldsymbol{\theta})\right)^{-1} \left(z_j(\boldsymbol{x}) - \mathbb{E}\left(\mathcal{GP}_j\left(\boldsymbol{x}, \boldsymbol{\theta}\right)\right)\right) \end{aligned} \tag{6.5}$$

$$\begin{aligned} I_{multi}(\boldsymbol{x}, \boldsymbol{\theta}) = &\left(z_j(\boldsymbol{x}) - \mathbb{E}\left(\mathcal{GP}_j\left(\boldsymbol{x}, \boldsymbol{\theta}\right)\right)\right)^{\mathsf{T}} \\ &\left(V_{o,j} + V_{m,j} + V_{c,j}(\boldsymbol{x}, \boldsymbol{\theta})\right)^{-1} \left(z_j(\boldsymbol{x}) - \mathbb{E}\left(\mathcal{GP}_j\left(\boldsymbol{x}, \boldsymbol{\theta}\right)\right)\right) \end{aligned} \tag{6.6}$$

In order to decide which parts of the parameter space to exclude a decision should be made based on the implausibility metric, often taking the form of a threshold T. Large implausibilities (for each formulation) indicate a parameter set was very unlikely to have produced an output that matched the observational data, given the included uncertainties. A rejection criteria can be formed for a particular parameter combination $\boldsymbol{\theta}$ as in Eq. (6.7).

$$I(\boldsymbol{\theta}) \begin{cases} \leq T & \text{if } \boldsymbol{\theta} \in \boldsymbol{\theta}_{nI}, \\ > T & \text{if } \boldsymbol{\theta} \in \boldsymbol{\theta}_{I} \end{cases} \tag{6.7}$$

The threshold value depends on the type of implausibility metric being considered. Andrianakis et al. state that a sensible threshold T for single $I_j(\boldsymbol{x}, \boldsymbol{\theta})$ or maximum $I_{max}(\boldsymbol{\theta})$ implausibilities (where the maximum is of a single implausibility set) can be determined by Pukelsheim's 3σ rule [71]. The rule states that any continuous unimodal distribution will contain at least 99.5% of probability mass within three standard deviations away from the mean [159]. For multivariate implausibilities the threshold T can be set as a high percentile ($\alpha > 95\%$) from a chi-squared distribution with either j, or the input size of $\boldsymbol{x}$, degrees of freedom [71], i.e. $T = F^{-1}_{\chi^2}(\alpha)$ the output from a chi-squared quantile function (inverse CDF). This can be thought of as performing a frequentist hypothesis test on the parameter combination, using a chi-squared (χ^2) test.

Furthermore, the algorithm requires a method for sampling the parameter space in order to assess the criteria. A simple approach is to draw samples from a uniform distribution bounded by the initial parameter domain. This works effectively with a LHD based approach. In this scenario the initial parameter space bounds are used, in conjunction with a simulator budget, to construct a LHD — here GMLHD from Section 4.2.3 are implemented. An emulator is constructed from the simulator runs and its output assessed at parameter combinations sampled from a uniform distribution where the bounds are from the parameter domain. A set of these sample parameters can then be rejected based on the given metric and criteria, and the bounds of the non-implausible region determined. A new wave can then be run with a LHD constructed from the new bounds.

Finally, a stopping criteria is constructed, based on two outcomes; all the space is deemed implausible or the emulator variance in the non-implausible region is less than the remaining uncertainties, i.e. $V_{c,j}(\boldsymbol{x}, \boldsymbol{\theta}_{nI}) < V_{o,j} + V_{m,j}$, which indicates

Algorithm 5 Bayesian History Matching for Wave k

$\boldsymbol{\theta}^k \sim$ GMLHC ▷ Draw parameters from GMLHC
$\boldsymbol{y}^k = \eta(\boldsymbol{x}, \boldsymbol{\theta}^k)$ ▷ Run the simulator at parameters
Draw n samples $\boldsymbol{\theta}_s^k \sim \mathcal{U}\left(\min(\boldsymbol{\theta}^k), \max(\boldsymbol{\theta}^k)\right)$ ▷ Sample parameter space
for $j = 1 :$ no. of outputs **do**
  Train and validate $\mathcal{GP}_j\left(\boldsymbol{x}, \boldsymbol{\theta}^k\right)$ ▷ Train and validate emulators
  $\left[\mathbb{E}\left(\mathcal{GP}_j\left(\boldsymbol{x}, \boldsymbol{\theta}_s^k\right)\right), V_{c,j}(\boldsymbol{x}, \boldsymbol{\theta}_s^k)\right] = \mathcal{GP}_j\left(\boldsymbol{x}, \boldsymbol{\theta}_s^k\right)$ ▷ Predictions at n samples of $\boldsymbol{\theta}^k$
  Calculate $I_j(\boldsymbol{x}, \boldsymbol{\theta}_s^k)$ ▷ Assess implausibility of samples
end for
Calcualate $I_{max}(\boldsymbol{\theta}_s^k)$
for $m = 1 : n$ **do**
  if $I_{max}(\boldsymbol{\theta}_{s,m}^k) < T$ **then**
    $\boldsymbol{\theta}_{nI}^k = \boldsymbol{\theta}_{s,m}^k$ ▷ Keep non-implausible samples
  end if
end for
bounds $= [\min(\boldsymbol{\theta}_{nI}^k), \max(\boldsymbol{\theta}_{nI}^k)]$ ▷ Obtain new GMLHC bounds
if any $(V_{c,j}^k(\boldsymbol{x}, \boldsymbol{\theta}) < (V_{o,j} + V_{m,j}))$ **or** isempty$(\boldsymbol{\theta}_{nI}^k)$ **then**
  Stop ▷ Stop if stopping criteria are met
end if

that the emulator is at least as certain about its predictions as the modeller is with the uncertainties due to model discrepancy and observation variability. The stated approach to BHM can be defined in Algorithm 5.

To illustrate BHM Algorithm 5 is applied to a simple numerical example (where the sampling stage is replaced with a uniform grid). In the example a simulator constructed from Eq. (6.8) models the experimental observation z, which is obtained from the 'true' process with noise, stated in Eq. (6.9); where $e \sim \mathcal{N}(0, 0.05)$. The observation $z(0.9) = 3.39$ has observational and model discrepancy uncertainties, $V_o = 0.05$ and $V_m = 0.04$ (estimated from the residual variance $\mathbb{V}((z - e) - y)$).

$$y = \eta(\boldsymbol{\theta}) = 5.5\left(0.15\cos(2\pi \times 0.75\boldsymbol{\theta}) + 1.25\sin(2\pi \times 0.1\boldsymbol{\theta})\right) \tag{6.8}$$

$$z(\boldsymbol{\theta}) = y(\boldsymbol{\theta}) - 0.3\sin(2\pi \times 0.15\boldsymbol{\theta}) + e \tag{6.9}$$

Figure 6.1 presents the experiential data point $z(0.9) = 3.39$ with $\pm\sqrt{V_o}$ intervals (shaded region) against the simulator and bias corrected outputs (i.e. $z - e$) across the parameter space $\boldsymbol{\theta}_s = \{-0.5, 0.005, \ldots, 5\}$ where a budget of four simulator

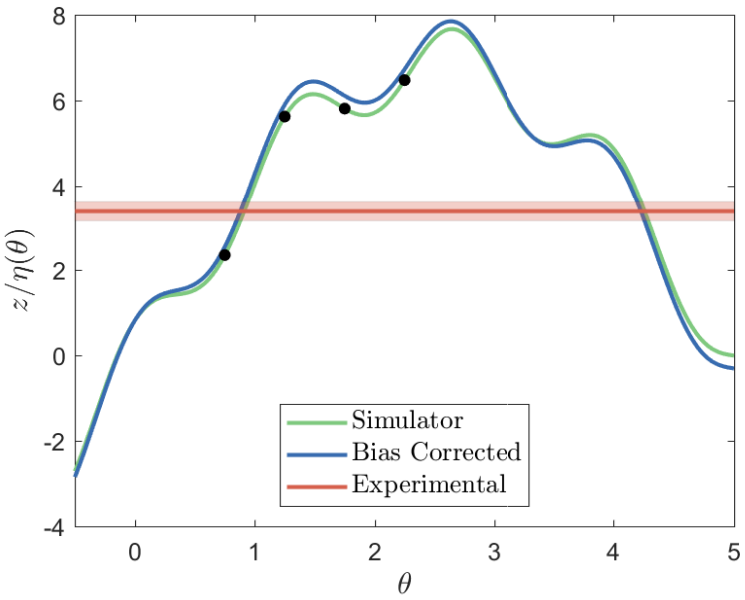


Figure 6.1: Simulator, model discrepancy and observational data (where the shaded region is $\pm\sqrt{V_o + V_m}$) for BHM numerical example. Where the inital simulator runs are $(\cdot)$.

evaluations have been performed in a space-filling manner $\boldsymbol{\theta}^1 = \{0.75, 1.25, 1.75, 2.25\}$. The observation $z = 3.39$ can be formed from two parameter 0.90 and 4.23 indicated by the cross-over in Fig. 6.1.

BHM was performed following Algorithm 5 with a simulator evaluation budget of four (for each space-filled design in wave k) where the single implausibility metric $I(\boldsymbol{\theta})$ and threshold $T = 3$ are implemented. The emulator for each wave was constructed from a constant mean and SE covariance functions with $\nu = 1 \times 10^{-8}$. The first, second and fourth waves are shown in Fig. 6.2.

In the first wave (Fig. 6.2a) the emulator predictions are most uncertain outside of $\boldsymbol{\theta}^1$ leading to these regions being classified as non-implausible. It can also be seen that the initial known simulator runs are deemed implausible, which can be visually confirmed as they are not within the remaining uncertainty bounds $z \pm \sqrt{V_o + V_m}$. Between these known simulator runs the code uncertainty increases leading to the parameter, around 1 and 2, being classed as non-implausible. By the second wave (Fig. 6.2b) additional simulator runs mean that the code uncertainty in the [0.75 2.25] interval are reduced below the remaining uncertainties and all judged as implausible. Simulator runs at the parameter bounds pin the code uncertainty removing the

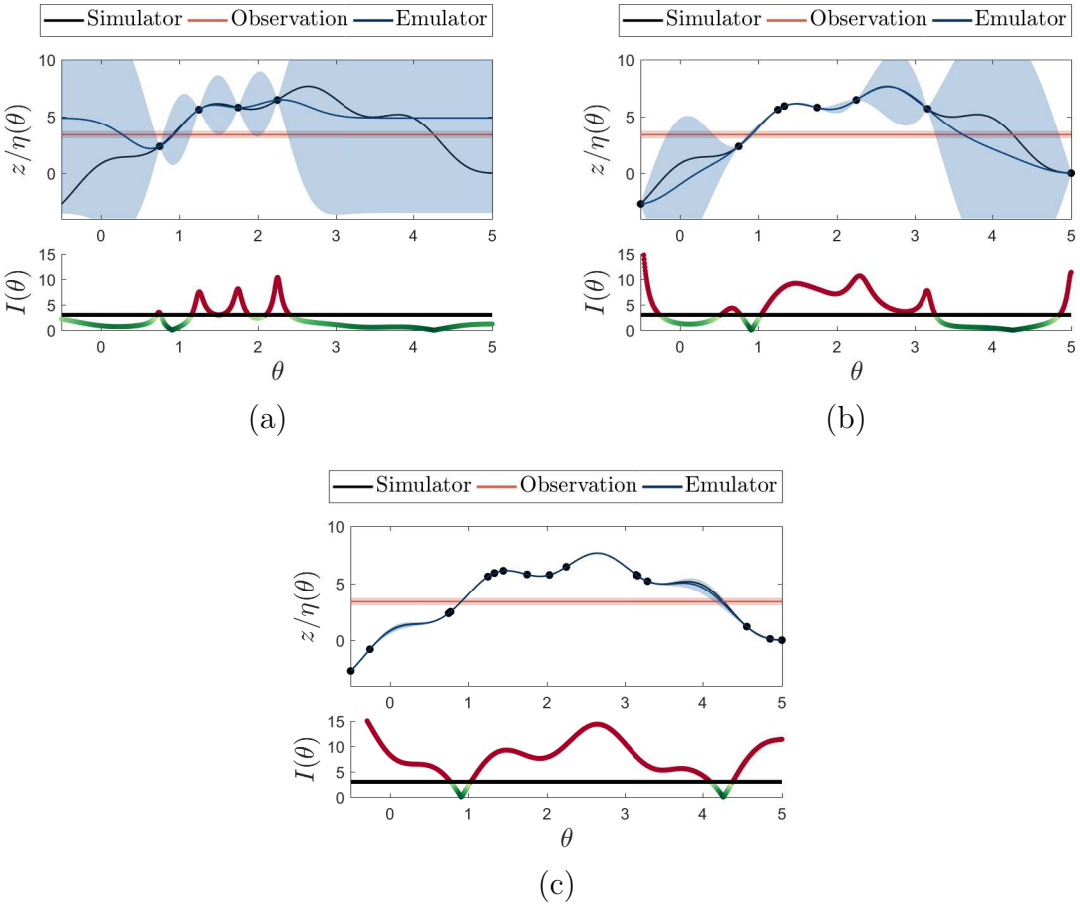


Figure 6.2: BHM waves $k = 1, 2, 4$ for the numerical example. Top panels show the observational data with $\pm\sqrt{V_o + V_m}$ shaded region against the simulator and emulator predictions (where the shaded regions indicates $\pm 3\sigma$), trained using the simulator runs $\eta(\boldsymbol{\theta}^k)$ $(\cdot)$. The bottom panels show the implausibility $I(\boldsymbol{\theta}_s^k)$ against the threshold $T = 3$, where green regions are non-implausible and red implausible. Panel (a), (b) and (c) show waves $k = 1, 2, 4$ respectively.

domain edges as implausible. By the final wave ($k = 4$) the code uncertainty has reduced across the space, and is lower than the remaining uncertainties in the non-implausible region. The non-implausible set $\boldsymbol{\theta}_{nI}$ at this wave clearly contain two regions around the solution 0.90 and 4.23.

Parameter Domain Sampling

BHM relies on sampling the parameter domain during each wave in order to evaluate the implausibility criterion. In Algorithm 5 a uniform sampling approach is suggested, however this may not be the most efficient method as samples may be wasted in space

that can be described as confidently implausible. Improvements to the techniques efficiency can be made by a more optimal approach to sampling the parameter domain, weighted to sample around the immediate non-implausible space.

One such approach is to calculate the non-implausible samples from the simulator evaluations (or samples from the previous wave) and to define a Gaussian distribution centred on these points where the variance is defined such that a small percentage are non-implausible [71]. By sampling N_s times from each of the Gaussian distributions, new parameter samples can be generated that should be sufficiently different from the old samples (as long as the variance is defined such that there are low non-implausibility rates). From these samples a set can be selected as the parameter samples for that wave based on a given simulator budget. Other proposed methods include evolutionary Monte Carlo aimed at producing uniform designs in subregions of the parameter space [160].

Alternatively the problem of where to sample in the parameter domain can be formed as a sequential process. This idea would involve defining a transition model between each wave using the non-implausible metric as an approximate likelihood. Subsequently a Sequential Monte Carlo (SMC) approach could be used to transfer information about where to sample in each wave via propagated samples based on their path directories, forming an SMC-BHM technique.

6.2.1 Approximate Posterior Sampling

The aim of applying BHM within forward model-driven SHM is to obtain calibrated parameters in a process that accounts for model discrepancy. In this application it is important that the posterior distributions of the parameters given observational data $p\left(\boldsymbol{\theta} \mid \boldsymbol{z}\right)$ are obtained. Importance sampling can be implemented at the end of the final wave as a method for obtaining an approximation to $p\left(\boldsymbol{\theta} \mid \boldsymbol{z}\right)$.

Importance sampling states that an unbiased estimate of the expectation integral can be formed form as shown in Eq. (6.10).

$$\mathbb{E}_p\left(f(x)\right) = \int f(x)p\left(x\right)\mathrm{d}x = \int q\left(x\right)\left(\frac{f(x)p\left(x\right)}{q\left(x\right)}\right)\mathrm{d}x = \mathbb{E}_q\left(\frac{f(X)p\left(X\right)}{q\left(X\right)}\right) \quad (6.10)$$

Where $X \sim q$ are independent draws from a proposal distribution. Given that X are discrete variables the expectation is equivalent to Eq. (6.11).

$$\mathbb{E}_p\left(f(x)\right) \approx \frac{1}{n}\sum_{i=1}^{n}\frac{f(X)p\left(X\right)}{q\left(X\right)} = \frac{1}{n}\sum_{i=1}^{n} f(X)w(X) \tag{6.11}$$

Where the ratio $p\left(X\right)/q\left(X\right) = w$, a set of importance weights.

When $p\left(x\right)$ is unknown but the unnormalised distribution is, i.e. $p^{un}(x) = Z_p\, p\left(x\right)$ (where $Z_p = \int p^{un}(x)\mathrm{d}x$ is the normalising constant), then importance sampling can be formed with an unnormalised proposal, i.e. $q^{un}(x) = Z_q\, q\left(x\right)$. In this scenario the estimator is as formed in Eqs. (6.12) and (6.13).

$$\mathbb{E}_p\left(f(x)\right) = \int q\left(x\right)\left(\frac{f(x)p\left(x\right)}{q\left(x\right)}\right)\mathrm{d}x = \frac{Z_q}{Z_p}\int q\left(x\right)\left(\frac{f(x)p^{un}(x)}{q^{un}(x)}\right)\mathrm{d}x \tag{6.12}$$

$$\mathbb{E}_p\left(f(x)\right) \approx \frac{Z_q}{Z_p}\frac{1}{n}\sum_{i=1}^{n}\frac{f(X)p^{un}(X)}{q^{un}(X)} = \frac{Z_q}{Z_p}\frac{1}{n}\sum_{i=1}^{n} f(X)w^{un}(X) \tag{6.13}$$

Where the unnormalised weights are $w^{un}(X) = p^{un}(X)/q^{un}(X)$ and $X \sim q^{un}$. The ratio of normalising constants Z_p/Z_q can also be approximated by importance sampling, as in Eq. (6.14) leading to Eq. (6.15).

$$\frac{Z_p}{Z_q} = \frac{1}{Z_q}\int p^{un}(x)\mathrm{d}x = \int q\left(x\right)\frac{p^{un}(x)}{q^{un}(x)}\mathrm{d}x \approx \frac{1}{n}\sum_{i=1}^{n} w^{un}(X) \tag{6.14}$$

$$\mathbb{E}_p\left(f(x)\right) \approx \frac{\frac{1}{n}\sum_{i=1}^{n} f(X)w^{un}(X)}{\frac{1}{n}\sum_{i=1}^{n} w^{un}(X)} \tag{6.15}$$

Using this form the technique can be applied to approximate a posterior density $p\left(\boldsymbol{\theta} \,|\, \boldsymbol{z}\right) = p\left(\boldsymbol{z} \,|\, \boldsymbol{\theta}\right)p\left(\boldsymbol{\theta}\right)/p\left(\boldsymbol{z}\right)$ when the evidence $p\left(\boldsymbol{z}\right)$ cannot be calculated. This requires setting $p^{un}(X) = p\left(\boldsymbol{z} \,|\, \boldsymbol{\theta}\right)p\left(\boldsymbol{\theta}\right)$ where $Z_p = p\left(\boldsymbol{z}\right)$, and the proposal distribution is $q^{un}(\boldsymbol{\theta}_q)$, leading to the approximation in Eq. (6.16).

$$p\left(\boldsymbol{\theta} \,|\, \boldsymbol{z}\right) \approx \frac{w^{un}(\boldsymbol{\theta}_q)}{\frac{1}{n}\sum_{i=1}^{n} w^{un}(\boldsymbol{\theta}_q)} \tag{6.16}$$

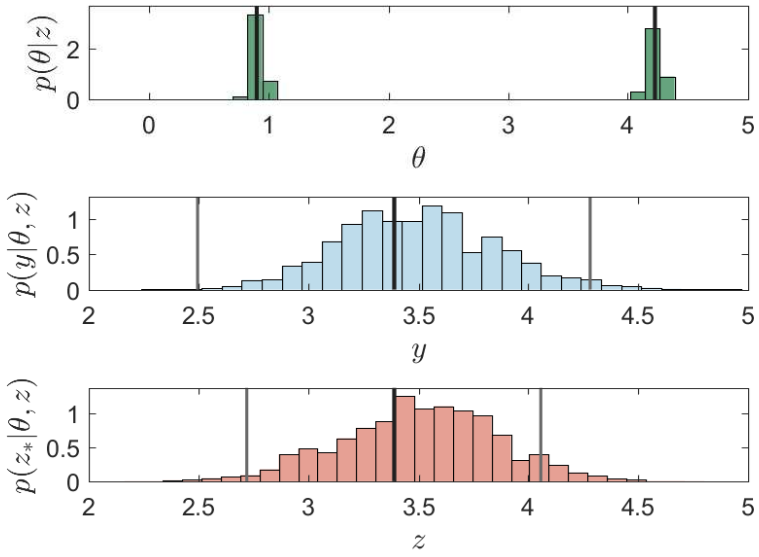


Figure 6.3: Posterior and predictive samples from a BHM numerical example. The top panel shows the approximate posterior $p(\boldsymbol{\theta} \mid z)$. The middle panel presents the simulator output $p(y \mid \boldsymbol{\theta}, z)$ given these posterior samples, where the black line denotes the 'true' value and the grey lines are $\pm 3(V_o + V_m)$. The bottom panel shows the bias corrected output $p(z_* \mid \boldsymbol{\theta}, z)$ (where $z_* = z - e$) given the posterior samples, where the black line denotes the 'true' value and the grey lines are $\pm 3V_o$.

Where $w^{un} = p(\boldsymbol{z} \mid \boldsymbol{\theta}_q) p(\boldsymbol{\theta}_q) / q^{un}(\boldsymbol{\theta}_q)$ is the probability of each sample $\boldsymbol{\theta}_q \sim q^{un}$. However, as the method does not involve a likelihood an approximation is formed as defined in Eq. (6.17), which is the product of multivariate Gaussian distributions over $z(\boldsymbol{x})$ for the set of inputs $\boldsymbol{x}$.

$$p(\boldsymbol{z} \mid \boldsymbol{\theta}) \approx L(\boldsymbol{\theta}) = \prod_{j=1}^{M} \mathcal{N}\left(\boldsymbol{z}(\boldsymbol{x}) \mid \mathbb{E}_j\left(\mathcal{GP}\left(\boldsymbol{x}, \boldsymbol{\theta}\right)\right), V_j(\boldsymbol{x}, \boldsymbol{\theta})\right) \tag{6.17}$$

Where $V_j(\boldsymbol{x}, \boldsymbol{\theta}) = V_{o,j} + V_{m,j} + V_{c,j}(\boldsymbol{x}, \boldsymbol{\theta})$ which assumes that these sources of uncertainty are normally distributed. As the emulator has a Student's t-distribution posterior this assumption means there are enough degrees of freedom for it to be approximately Gaussian distributed. The proposal distribution can be formulated as a multivariate Gaussian distribution as presented in Eq. (6.18).

$$q^{un}(\boldsymbol{\theta}) = \mathcal{N}\left(\boldsymbol{\theta} \mid \mu_{nI}, \kappa \Sigma_{nI}\right) \tag{6.18}$$

Where μ_{nI} and Σ_{nI} are the sample mean and variance-covariance from the non-implausible set after the last wave and κ is an inflation parameter to ensure good coverage of the space.

The choice of prior $p(\boldsymbol{\theta})$ depends on the modellers beliefs from the last wave. However it is often reasonable to assume a constant prior over the final non-implausible set, as it is often a fraction of the original parameter domain. This means the weights in Eq. (6.16) become $w^{un} = L(\boldsymbol{\theta}_q)/q^{un}(\boldsymbol{\theta}_q)$ where $\boldsymbol{\theta}_q$ are a number of samples from q^{un} and the constant prior essentially truncates the proposal samples to be within the final non-implausible domain.

Lastly the approximate posterior from Eq. (6.16) can be re-sampled in order generate direct samples from the posterior. This involves drawing N_q samples where the probability of occurrence is defined by the normalised weights $w(\boldsymbol{\theta}_q) = w^{un}(\boldsymbol{\theta}_q)/\sum w^{un}(\boldsymbol{\theta}_q)$.

Figure 6.3 demonstrates importance sampling and re-sampling on from the numerical example in Section 6.2 where $N_q = 10,000$ and $\kappa = 2$. The re-sampled posterior samples are subsequently used to draw Monte Carlo realisations of the simulator and bias corrected output. The results show that the emulator has been adequately calibrated with the two parameter solutions lying within the central probability mass. Furthermore the simulator and bias corrected results lie within the given uncertainty bounds.

6.2.2 Sequential Based Approaches

Central to implementing BHM is generating and evaluating computer DoEs. These provide the information required to construct emulators with which to assess and classify the parameter domain in a computationally efficient manner. As a result alternative DoE formulations can be used, as opposed to space-filled designs such as the Generalised Maximum Latin Hypercube (GMLHC). Two heuristic sequential based methods are explored with a view to move towards information based DoEs. Two metrics, probability of non-implausibility and expected (un)improvement, adapted from the field of Bayesian optimisation, provide criteria for selecting new simulator evaluations in a sequential manner and are explored in the following sections.

Probability of Non-implausibility

Probability of non-implausibility assesses the chance of a parameter combination being non-implausible given the observation and system uncertainties [108]. Mathematically this is the probability that $\boldsymbol{\theta} \in \boldsymbol{\theta}_{nI}$ if the mean prediction from the emulator lies within the uncertainty bounds, $D_{-,j}(\boldsymbol{x}) \leq \mathbb{E}_j\left(\mathcal{GP}\left(\boldsymbol{x}, \boldsymbol{\theta}\right)\right) \leq D_{+,j}(\boldsymbol{x})$ as defined for the ith parameter combination Eq. (6.19).

$$p\left(\boldsymbol{\theta}_i \in \boldsymbol{\theta}_{nI}\right) = \Phi\left(\frac{D_{+,j}(\boldsymbol{x}) - \mathbb{E}\left(\mathcal{GP}_j\left(\boldsymbol{x}, \boldsymbol{\theta}_i\right)\right)}{V_{c,j}(\boldsymbol{x}, \boldsymbol{\theta}_i)^{-0.5}}\right) - \Phi\left(\frac{D_{-,j}(\boldsymbol{x}) - \mathbb{E}\left(\mathcal{GP}_j\left(\boldsymbol{x}, \boldsymbol{\theta}_i\right)\right)}{V_{c,j}(\boldsymbol{x}, \boldsymbol{\theta}_i)^{-0.5}}\right) \tag{6.19}$$

Where $D_{+,j}(\boldsymbol{x})$ and $D_{-,j}(\boldsymbol{x})$ are the upper and lower non-implausible output bounds $z_j(\boldsymbol{x}) \pm v_s\sqrt{V_{o,j} + V_{m,j}}$ with v_s defining the bound width, and $\Phi\left(\cdots\right)$ a standard Gaussian CDF. This variance scalar effectively behaves as the threshold in the implausibility metric and here is set as 3 due to Pukelsheim's 3σ rule.

The probability of non-implausibility is similar to the probability of improvement used in Bayesian optimisation. This heuristic when implemented in Bayesian optimisation is used to determine the probability of improving on the current minimum across a space [130]. In contrast the formulation in Eq. (6.19) seeks parameter combinations that are likely to be within the output bounds $[D_{+,j}(\boldsymbol{x})\ D_{-,j}(\boldsymbol{x})]$, leading to the confident exclusion of parameter regions when the probability of being non-implausible is close to zero and the reverse when probability is close to one. The non-implausibility criteria is therefore defined as parameter combinations where $p\left(\boldsymbol{\theta}_i \in \boldsymbol{\theta}_{nI}\right) = 1$.

In sequential BHM each wave seeks to find the parameter combination with the largest probability less than one and to use this set as the next simulator evaluation. This reflects the belief that probability one states — with certainty given the bounds — that the parameter set output matches the output bounds, where the largest probability less than one (and greater than zero) will indicate a potential match which could be made certain either way by improving the code uncertainty of emulator prediction for that set. A stopping criteria can be formed similar to Algorithm 5 where the process stops when the code uncertainty of the parameters with probability greater than zero is less than the observational and model discrepancy uncertainties.

Figure 6.4 demonstrates a selection of waves when probability of non-implausibility

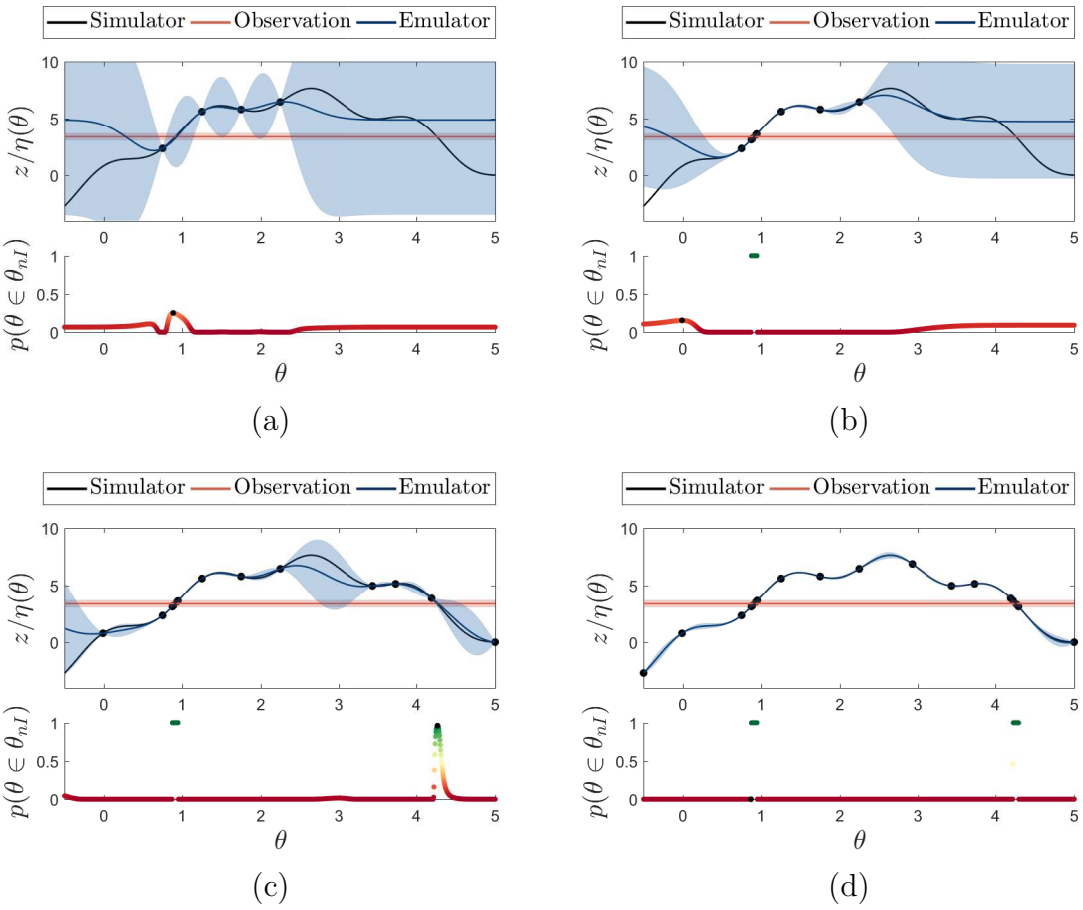


Figure 6.4: Sequential BHM using probability of non-implausibility for waves $k = 1, 5, 10, 18$ for the numerical example. Top panels show the observational data with $\pm\sqrt{V_o + V_m}$ shaded region against the simulator and emulator predictions (where the shaded regions indicated $\pm 3\sigma$), trained using the simulator runs $\eta(\boldsymbol{\theta}^k)$ $(\cdot)$. The bottom panels shows the probability of non-implausibility $p\left(\theta \in \boldsymbol{\theta}_{nI}\right)$, where $(\cdot)$ indicates the new simulator evaluation for the $(k+1)$th wave. Panel (a), (b), (c) and (d) show waves $k = 1, 5, 10, 18$ respectively.

is implemented as part of a sequential BHM approach for the numerical example in Fig. 6.1; with the same emulator mean and covariance functions and uncertainties. Between waves 1 and 5 (Fig. 6.4a and Fig. 6.4b) it can be seen that the algorithm spends simulator evaluations exploiting the nearby non-implausible region, with the next simulator evaluation for wave 6 being away from this area. The algorithm becomes more exploratory between waves 5 and 10, where the second non-implausible region is starting to be identified. Finally by wave 18 the stopping criteria has been met and the two non-implausible regions have been found. The approach requires more simulator evaluations, 22, than Algorithm 5, 16. This is due to the probability of non-implausibility being a highly exploitative criteria, as shown by the numerous

evaluations about the non-implausible regions.

By deriving the probability of non-implausibility, BHM can be defined as a sub-category of ABC [108]. Essentially this formulation becomes ABC with a uniform prior $p(\boldsymbol{\theta}) \propto \mathbb{1}_{\boldsymbol{\theta} \in \boldsymbol{\Theta}}$ over the assessed parameter domain $\boldsymbol{\Theta}$ and an acceptance kernel $\mathbb{1}_{\eta(\theta) \in [D_{+,j}(\boldsymbol{x})\ D_{-,j}(\boldsymbol{x})]}$ meaning the approximate posterior becomes,

$$p(\boldsymbol{\theta} \mid \boldsymbol{z}) \propto \begin{cases} 1 & \text{if } \boldsymbol{\theta} \in \boldsymbol{\theta}_{nI}, \\ 0 & \text{otherwise} \end{cases},$$

where a posterior probability of zero means an implausible parameter combination. This comparison allows BHM to gain useful properties from ABC such as that ABC performs exact inference under uniform additive model discrepancy [161].

Expected (un)Improvement

Another heuristic with an improved balance between exploratory and exploitative objectives is expected (un)improvement. This proposed sequential design criteria is a development and reformulation of expected improvement utilised in Bayesian optimisation [162] combining the probability of matching observations within the uncertainty bounds with the expected magnitude of the improvement at a particular parameter combination.

To construct the criteria, (un)improvement must be defined; where improvement is typically $I(\boldsymbol{\theta}) = \max(f_{min} - \eta(\boldsymbol{\theta}), 0)$ in Bayesian optimisation [162]. This definition states that an improvement occurs when the simulator prediction is less than the current function minimum, with the improvement being zero when the simulator prediction is lower. In a BHM context the function minimum f_{min} is replaced by the observation with its defined uncertainty bounds. In this context the notion of improvement is not what is required, instead the search is for the 'smallest improvement' from the known bounded observations. In addition there are two improvement criteria as the observation is upper and lower bounded. This leads to the formulation of a criteria that will be zero or positive when a parameter is within the observation bounds and negative for the reverse. This sequential criteria is designed from taking the expectation of two (un)improvement criteria, where an (un)improvement occurs when the expected emulator prediction is below the

lower bound $I_{lb}(\boldsymbol{\theta}) = \max(D_- - \mathbb{E}\left(\mathcal{GP}(\boldsymbol{\theta})\right), 0)$ or greater than the upper bound $I_{ub}(\boldsymbol{\theta}) = \max(\mathbb{E}\left(\mathcal{GP}(\boldsymbol{\theta})\right) - D_+, 0)$. The expected (un)improvement for all possible emulator values at a parameter combination is found by taking the expectation, which can be calculated in closed form for the lower and upper bounds in Eqs. (6.20) and (6.21) respectively.

$$\mathbb{E}_{\eta\sim\mathcal{GP}(\boldsymbol{\theta})}\left(I_{lb}(\boldsymbol{\theta})\right) = \sqrt{V_c(\boldsymbol{\theta})}\left(\gamma_{lb}\Phi\left(\gamma_{lb}\right) + \Phi\left(\gamma_{lb}\right)\right) \tag{6.20}$$

$$\mathbb{E}_{\eta\sim\mathcal{GP}(\boldsymbol{\theta})}\left(I_{ub}(\boldsymbol{\theta})\right) = \sqrt{V_c(\boldsymbol{\theta})}\left(-\gamma_{ub}\Phi\left(-\gamma_{ub}\right) + \Phi\left(\gamma_{ub}\right)\right) \tag{6.21}$$

Where $\gamma_{lb} = (D_- - \mathbb{E}\left(\mathcal{GP}(\boldsymbol{\theta})\right))/\sqrt{V_c(\boldsymbol{\theta})}$ and $\gamma_{ub} = (D_+ - \mathbb{E}\left(\mathcal{GP}(\boldsymbol{\theta})\right))/\sqrt{V_c(\boldsymbol{\theta})}$ are the standardised distances between the bounds and mean emulator prediction. The expected (un)improvement criteria is the negative sum of Eqs. (6.20) and (6.21) as defined in Eq. (6.22) and takes the same units as the emulator output.

$$-EI(\theta) = -(\mathbb{E}_{\eta\sim\mathcal{GP}(\boldsymbol{\theta})}\left(I_{lb}(\boldsymbol{\theta})\right) + \mathbb{E}_{\eta\sim\mathcal{GP}(\boldsymbol{\theta})}\left(I_{ub}(\boldsymbol{\theta})\right)) \tag{6.22}$$

The criteria can be combined with probability of non-implausibility to form a sequential BHM algorithm, where the approach follows that outlined previously (with the same non-implausibility and stopping criteria) with a different method for selecting new simulator evaluations. New runs are obtained for the parameter combination, with probability of non-implausibility less than one, where the expected (un)improvement ($-EI(\theta)$) is maximum.

Figure 6.5 presents a selection of waves from performing sequential BHM using expected (un)improvement for the numerical example with the same emulator mean and covariance functions and uncertainties. By wave 3 the method has begun exploring the parameter space with simulator evaluations concentrated at the observation bounds, as with probability of non-implausibility. Wave 10 demonstrates that the approach has explored the parameter space and begins to exploit locations that are likely to be plausible. At iteration 14 the algorithm has met the stopping criteria showing a greater efficiency than both probability of non-implausible and the space-filling approaches.

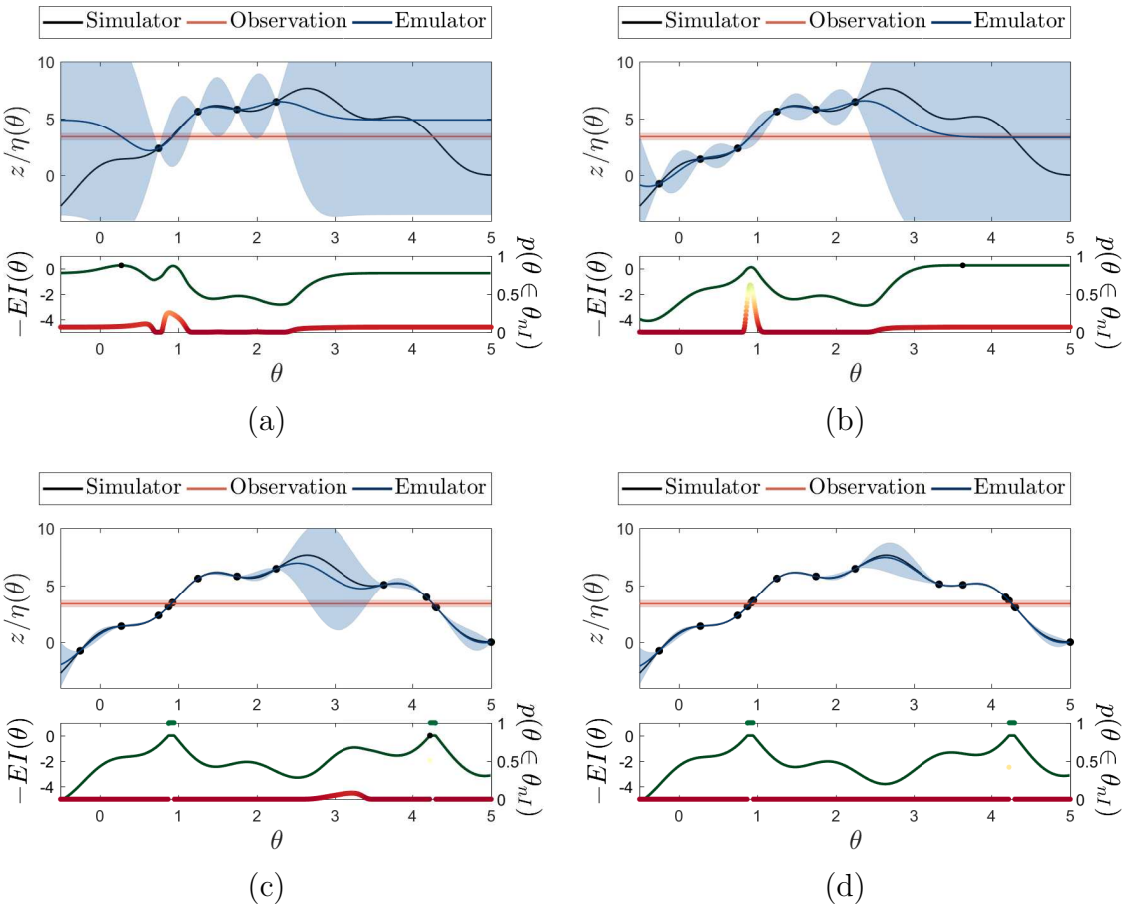


Figure 6.5: Sequential BHM using probability of non-implausibility for waves $k = 1, 3, 10, 14$ for the numerical example. Top panels show the observational data with $\pm\sqrt{V_o + V_m}$ shaded region against the simulator and emulator predictions (where the shaded regions indicated $\pm 3\sigma$), trained using the simulator runs $\eta(\boldsymbol{\theta}^k)$ $(\cdot)$. The bottom panels shows the negative expected (un)improvement $-EI(\boldsymbol{\theta})$ and probability of non-implausibility $p\left(\theta \in \boldsymbol{\theta}_{nI}\right)$, where $(\cdot)$ indicates the new simulator evaluation for the $(k+1)$th wave. Panel (a), (b), (c) and (d) show waves $k = 1, 3, 10, 14$ respectively.

Information Based Approaches

Information based approaches are alternatives to the aforementioned heuristics within Bayesian optimisation. These techniques seek to design criteria from information theory that maximises the expected information gain on the GP posterior. One such approach, Entropy Search (ES), uses the current information, quantified by the negative differential entropy of $p\left(\boldsymbol{\theta}_s^k \,|\, \mathcal{D}_k\right)$, to select a new point $\boldsymbol{\theta}^{k+1}$ that will minimise the expected negative differential entropy, where the sequential criteria is defined in Eq. (6.23) [129, 163].

$$ES(\boldsymbol{\theta}_s^k) = H\left(p\left(\boldsymbol{\theta}_s^k \mid \mathcal{D}_k\right)\right) - \mathbb{E}_{p(\boldsymbol{\eta} \mid \mathcal{D}_k)}\left(p\left(\boldsymbol{\theta}_s^k \mid \mathcal{D}_k \cup \{\boldsymbol{\theta}, \boldsymbol{\eta}\}\right)\right) \tag{6.23}$$

Where $\mathcal{D}_k$ is the set of current known simulator outputs $\boldsymbol{\eta}$ for a given set of parameter combinations $\boldsymbol{\theta}$, and $H(p(\boldsymbol{\theta})) = -\int p(\boldsymbol{\theta}) \log p(\boldsymbol{\theta}) \mathrm{d}\boldsymbol{\theta}$, the negative differential entropy, which for BHM is defined as in Eq. (6.24).

$$\begin{aligned} H(p(\boldsymbol{\theta}_i \in \boldsymbol{\theta}_{nI})) = -\int & p(\boldsymbol{\theta}_i \in \boldsymbol{\theta}_{nI}) \log p(\boldsymbol{\theta}_i \in \boldsymbol{\theta}_{nI}) \\ & - (1 - p(\boldsymbol{\theta}_i \in \boldsymbol{\theta}_{nI})) \log(1 - p(\boldsymbol{\theta}_i \in \boldsymbol{\theta}_{nI})) \mathrm{d}\boldsymbol{\theta} \end{aligned} \tag{6.24}$$

Equation (6.23) in practice proves demanding to evaluate as the entropy does not have an analytical solution and the distribution $p\left(\boldsymbol{\theta}_s^k \mid \mathcal{D}_k \cup \{\boldsymbol{\theta}, \boldsymbol{\eta}\}\right)$ must be calculated for numerous combinations of $\boldsymbol{\theta}$ and $\boldsymbol{\eta}$.

In contrast to ES, Predictive Entropy Search (PES) targets the mutual information between $\boldsymbol{\theta}_s^k$ and $\boldsymbol{\eta}$ given $\mathcal{D}_k$ leading to a sequential design criteria defined in Eq. (6.25) [164].

$$PES(\boldsymbol{\theta}_s^k) = H(p(\boldsymbol{\eta} \mid \mathcal{D}_k, \boldsymbol{\theta})) - \mathbb{E}_{p\left(\boldsymbol{\theta}_s^k \mid \mathcal{D}_k\right)}\left(p\left(\boldsymbol{\eta} \mid \mathcal{D}_k, \boldsymbol{\theta}, \boldsymbol{\theta}_s^k\right)\right) \tag{6.25}$$

This formulation leads to calculating posterior distributions (and their entropies) which for a GP have analytical forms or can be approximated more easily, simplifying the sequential design process. These approaches are likely to improve the efficiency and effectiveness of sequential BHM and are left as areas of further research.

6.2.3 Model Discrepancy

BHM accounts for model discrepancy by defining a prior variance V_m, stating an assumption of uniform additive discrepancy across the space. As stated in Section 6.2.2, BHM is a subcategory of ABC and therefore has the property of performing exact Monte Carlo inference for a uniform additive model discrepancy. In order to illustrate this result a numerical example is outlined.

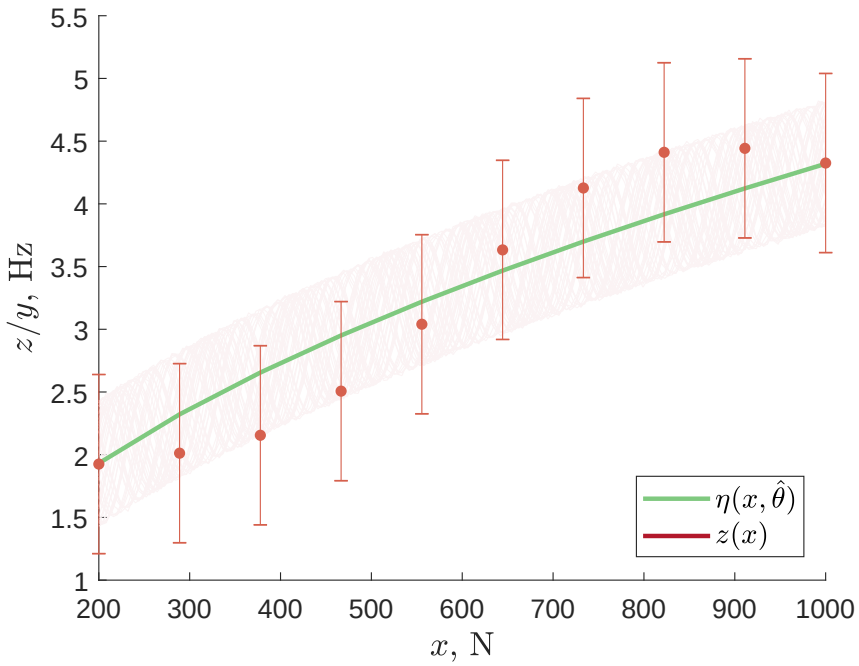


Figure 6.6: BHM model discrepancy numerical example where the red $(\cdot)$ are the observational data with $\pm\sqrt{V_o + V_m}$ bounds and the red $(-)$ realisations from Eq. (6.27).

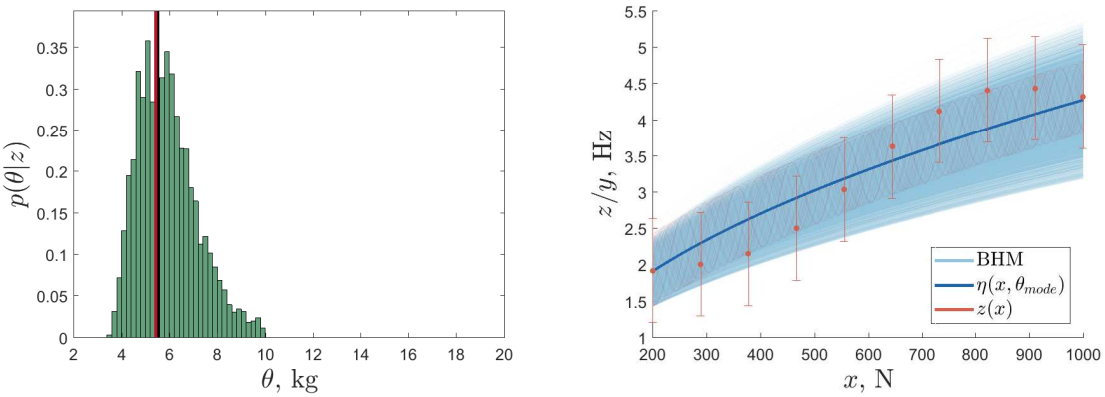
A simulator is constructed from a mass, tensioned wire system from Section 2.1.3 redefined in Eq. (6.26), where M is mass, T is tension $l = 1\text{m}$ is the length and w_n is the natural frequency.

$$\eta(x, \theta) = w_n(T, M) = \frac{1}{\pi}\sqrt{\frac{T}{Ml}} \tag{6.26}$$

In the example model discrepancy is considered additive and sinusoidal, i.e. $\delta(x) = 0.5\sin(2\pi \times 0.01x + \phi)$ where $\phi \sim \mathcal{U}(0, 2\pi)$ is a random phase. The observational process is defined in Eq. (6.27) with a 'true' mass $\hat{\theta} = 5.43\text{kg}$ and observational uncertainty $e \sim \mathcal{N}(0, 0.01^2)$. A comparison of the simulator and experimental data is displayed in Fig. 6.6.

$$z(x) = \eta(x, 5.43) + \delta(x) + e \tag{6.27}$$

The uncertainties used in BHM are $V_o = 0.01^2$ from the noise and $V_m = 0.5$ due to the maximum and minimum of the discrepancy $\delta(x)$. A multivariate implausibility metric is implemented with a threshold calculated from the 99% quantile from a 10



(a) (b)

Figure 6.7: BHM model discrepancy numerical example. Panel (a) shows the approximate posterior over the parameters from importance sampling when $\kappa = 2$; the black (−) is the modal estimate and the red (−) the 'true' parameter value. Panel (b) presents Monte Carlo realisations of the simulator output given the parameter posterior where the blue (−) is the modal estimate, the red (·) are the observational data with $\pm\sqrt{V_o + V_m}$ bounds and the red (−) realisations from Eq. (6.27).

degree of freedom χ^2-distribution. The emulator is constructed from a linear mean function $m(\boldsymbol{x}, \boldsymbol{\theta}) = [\boldsymbol{x}, \boldsymbol{\theta}]^\mathsf{T}\boldsymbol{\beta}$ and Matérn covariance (where $p = 2$) and a nugget term $\nu = 1 \times 10^{-8}$. A non-sequential approach is used where the parameter domain is uniformly sampled with $50,000$ samples. The parameter domain bounds were [2 20]kg and the experimental data was obtained at 10 equally space points from 200-1000N when $\phi = 0$.

BHM reaches the stopping criteria after one wave and the approximate posterior from importance sampling is presented in Fig. 6.7a along with Monte Carlo realisations of the simulator output in Fig. 6.7b. It can be seen that the 'true' parameter value $\hat{\theta} = 5.43$ is within the central probability mass with a modal estimate being $\theta_{mode} = 5.53$ showing good agreement.

Another scenario of interest is when the model discrepancy is not a sum. This may occur in most practical engineering scenarios, where the missing physics are coupled with the known physics. In this scenario BHM should not be expected to perform exact inference, but will result in an inflated parameter posterior where there should be a portion of probability mass where the 'true' parameter occurs. In order to demonstrate this scenario a numerical example using the mass, tensioned wire system is demonstrated — as shown in Fig. 6.8. Here the observational process is an offset mass, tensioned wire system defined as in Eq. (6.28); where $a = 0.2$ and $b = 1 - 0.2$.

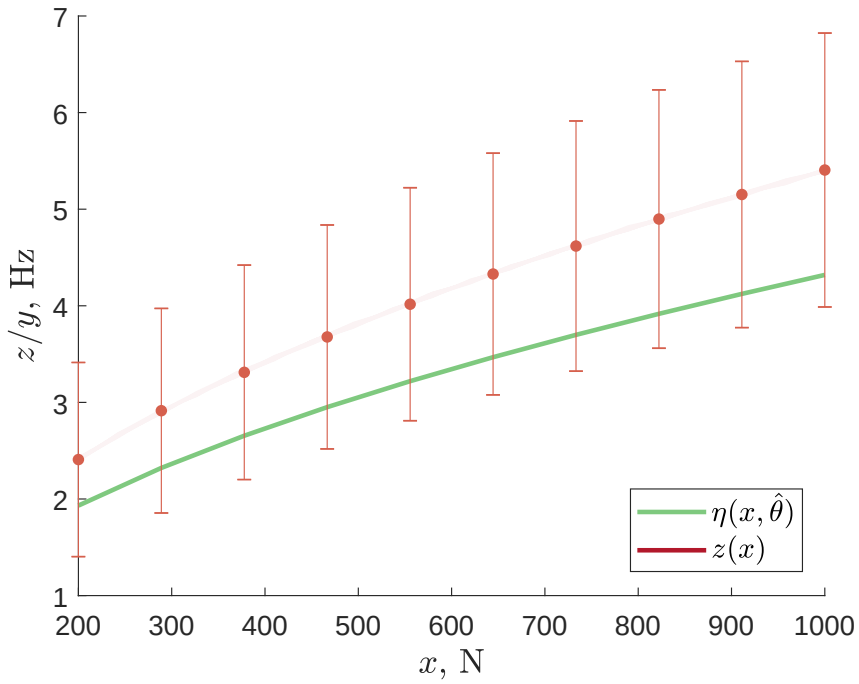
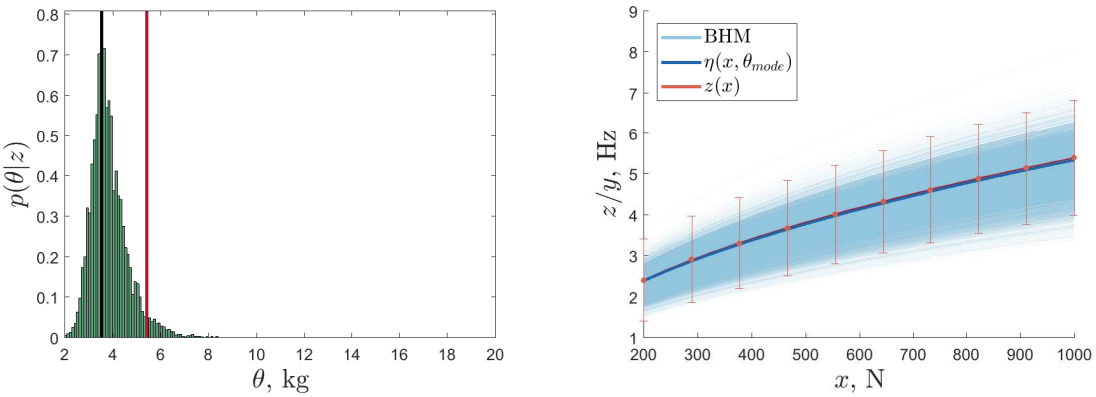


Figure 6.8: BHM model discrepancy numerical example with coupled discrepancy, where the red $(\cdot)$ are the observational data with $\pm\sqrt{V_o + V_m(\boldsymbol{x})}$ bounds and the red $(-)$ realisations from Eq. (6.27).

$$z(x) = \frac{1}{2\pi}\sqrt{\frac{T(a+b)}{M(ab)}} + e \tag{6.28}$$

In this example the same settings are used as the previous example, however the model discrepancy uncertainty is defined as $V_m(\boldsymbol{x}) = [1, 1.11, \ldots, 2]$, describing a linear increase in the model discrepancy. The offset produced by the model discrepancy in Eq. (6.28) will affect the ability of the BHM process to approximate the posterior parameter distribution. This is because the calibration process is still limited to the incorrect functional form defined by the simulator. In addition the offset will cause a bias in the posterior parameter distribution as is shown in Fig. 6.9a. Although the parameter posterior distribution contains probability mass at the 'true' parameter value there is a significant discrepancy between the modal estimate $\theta_{mode} = 3.54$ and the 'true' parameter value $\hat{\theta} = 5.43$. Furthermore it can be seen in Fig. 6.9b that the modal parameter solution produces an output that closes matches the observational data, with a NMSE of 0.17. This indicates that the method will try to calibrate the simulator given the modelling assumptions of a model discrepancy that is additive.

In contrast, the result in Fig. 6.9a shows that the 'true' parameter is within the



(a) (b)

Figure 6.9: BHM model discrepancy numerical example with coupled discrepancy. Panel (a) show the approximate posterior over the parameters from importance sampling when $\kappa = 2$; the black $(-)$ is the modal estimate and the red $(-)$ the 'true' parameter value. Panel (b) presents Monte Carlo realisations of the simulator output given the parameter posterior where the blue $(-)$ is the modal estimate, the red $(\cdot)$ are the observational data with $\pm\sqrt{V_o + V_m(\boldsymbol{x})}$ bounds and the red $(-)$ realisations from Eq. (6.27).

probability mass, and given that in most real applications the model discrepancy is completely unknown, BHM can be a practical tool given the modeller limited knowledge.

6.3 Representative Five Storey Building Case Study

Calibration of five bending modes of a representative five storey building structure was performed using BHM in order to demonstrate the approaches applicability for forward model-driven SHM. Modal testing was performed on a representative five storey building structure made from aluminium 6082 under different pseudo-damage extents as shown in Fig. 5.10. These pseudo-damage extents were added masses $m = \{0, 0.1, \ldots, 0.5\}$kg fixed to the first floor of the structure demonstrated in Fig. 5.10b. The structure was excited with a 409.6Hz bandwidth Gaussian noise via an electrodynamic shaker, with sample rate and sample time chosen to allow a frequency resolution of 0.05Hz. Accelerometers were placed at each of the five floors in order to obtain the first five bending modes. 40 averages were acquired for each measurement and ten repeats were performed for each damage extent in order to obtain an understanding of the underlying modal frequency distributions.

Parameter		Lower Bound	Upper Bound
Elastic Modulus	E	63.9GPa	78.1GPa
Poisson's Ratio	ν	0.297	0.363
Density	ρ	2493kg/m^3	3047kg/m^3

Table 6.1: The prior parameter bounds for BHM on the five storey representative building structure.

The observational data $z(\boldsymbol{x}_z)$ used within the calibration process were the mean natural frequencies when $\boldsymbol{x}_z = \{0, 0.3, 0.5\}$kg. The unseen validation set were the full repeat measurements of $z(\boldsymbol{x}_z)$ as well as those from the $\{0.1, 0.2, 0.4\}$kg pseudo-damage extents, with the inputs collectively denoted as $\boldsymbol{x}_*$. This highlights that with a small subset of damage data predictions can be made using BHM for forward model-driven SHM.

The simulator $\eta(\boldsymbol{x}, \boldsymbol{\theta})$ was a modal FEA model where the five bending natural frequencies were extracted as a set of outputs $\boldsymbol{y}$. Evaluations of the simulator were acquired for the six damage extents $\boldsymbol{x} = \{0, 0.1, \ldots, 0.5\}$kg and a range of parameter $\boldsymbol{\theta}$ values within a set of prior bounds; set as $\pm 10\%$ of typical material properties for aluminium 6082 as shown in Table 6.1. Simulator runs for parameter combinations determined by a fifty point, three dimensional GMLHC, were implemented as training data for five independent GP emulators with a separate ten point three dimensional GMLHC used to generate validation data.

6.3.1 Bayesian History Matching

Non-sequential BHM was implemented using GMLHCs to provide training data for five independent GP emulators. Each emulator was constructed from linear mean $m(\boldsymbol{x}, \boldsymbol{\theta}) = [\boldsymbol{x}, \boldsymbol{\theta}]^\mathsf{T}\boldsymbol{\beta}$ and Matérn (where $p = 2$) covariance functions with a nugget of $\nu = 1 \times 10^{-8}$. Exploration of the parameter domain was performed via $100,000$ samples from uniform distributions over the bounds. A multivariate implausibility (Eq. (6.5)) was implemented with the non-implausibility criteria being when the maximum multivariate implausibility for all five outputs (the five natural frequencies) was less than the 99% quantile for a three degree of freedom χ^2-distribution (reflecting the size of $\boldsymbol{x}_z$). The observational $V_{o,j}$ and model discrepancy $V_{m,j}$ uncertainties, set for the first BHM wave are displayed in Table 6.2 and were estimated from the experimental output variance of the ten repeats at the training inputs and from the modeller's judgement respectively. The stopping criteria required the

Uncertainty		ω_1	ω_2	ω_3	ω_4	ω_5
Observational	V_o	3×10^{-5}	0.02	0.09	0.05	0.01
Model Discrepancy	V_m	1.5	0.01	0.01	1	1

Table 6.2: The process uncertainties defined in the implausibility measure utilised for performing BHM on the five storey representative building structure.

code uncertainty from each of the five emulators to be less than their respective observational and model discrepancy uncertainties $V_{o,j} + V_{m,j}$.

The stopping criteria was met after one wave as the code uncertainty for each of the five emulators had an order of magnitude $\approx 10^{-4}$. This low level of code uncertainty indicates that the emulators had captured the simulator behaviour well, and the diagnostic checks from Section 4.2.2 evidenced that the emulators were valid. After the first wave a non-implausible space $\approx 2.3\%$ of the original space was identified.

In order to visualise the non-implausible space from the non-implausibility criteria minimum implausibility and optical depth plots were created. These quantities divide the parameter space into bins where each of the $100,000$ samples (from the uniform parameter domain sampling) are placed. Minimum implausibility takes the lowest value of implausibility below the threshold for the set of samples within a given bin. This provides an indication of which parts of the parameter space can be discarded irrespective of the other parameters. Optical depth is the ratio between non-implausible samples and the total number of samples within a given bin, providing an estimate of the probability of finding a non-implausible parameter combination given the set within a bin. Figure 6.10 presents these quantities after the first wave when each parameter is divided into thirty bins. Here it can be seen that high values of elastic modulus and low values of density are identified as non-implausible with Poisson's ratio being relatively insensitive to the outputs. There is a clear linear correlation between the non-implausible space of the elastic modulus and density, displayed in the bottom left and top right quadrants of Fig. 6.10.

As the stopping criteria has been met approximate posterior densities can be formed using importance sampling and re-sampling. A Gaussian proposal distribution with $\kappa = 2$ was used to generate $100,000$ samples with which to assess the normalised weights using the methodology presented in Section 6.2.1. $100,000$ samples were subsequently obtained by re-sampling the posterior distribution. Figure 6.11 presents the marginal and pairwise joint posterior distributions, which are visually similar to the minimum implausibility and optical depths; with a linear relationship between

low density and high elastic modulus values and a relatively insensitive effect from Poisson's ratio in the pairwise joint distributions. Figure 6.12 displays the marginal posterior distribution for each parameter, all showing bi-modal distributions. The pairwise joint posteriors indicate that these modes correspond to opposite ends of the marginal distribution, for example the elastic modulus modal value around 75GPa corresponds to density mode around 2500kg/m^3 and the two Poisson's ratio modes around 0.3 and 0.36. These results show the methods ability to account for multi-modal behaviour within the defined parameter domain.

The output distributions for each of the five natural frequencies were obtained via Monte Carlo sampling the posterior parameter distribution. $1,000$ samples were taken from the re-sampled parameter posterior distributions and propagated through each of the five emulators in order to obtain realisations of the output distributions. As the code uncertainty across all emulators was extremely low $\approx 10^{-4}$, each emulator mean was taken as deterministic. It is noted that if the emulator variances were not several orders of magnitude lower than the combined observational and model discrepancy uncertainties to be deemed negligible, posterior sampling of the GP should be implemented via Eq. (6.29).

$$\tilde{\eta}(\boldsymbol{x}_*, \tilde{\theta}) = \mathbb{E}\left(\mathcal{GP}\left(\boldsymbol{x}_*, \tilde{\theta}\right)\right) + R_{*,*}^{\mathsf{T}}\zeta \tag{6.29}$$

Where $\mathbb{E}\left(\mathcal{GP}\left(\boldsymbol{x}_*, \tilde{\theta}\right)\right)$ is the mean prediction from the GP emulator, $R_{*,*}$ is the upper matrix from the Cholesky decomposition of the predictive covariance $\text{cov}\left(\mathcal{GP}\left(\boldsymbol{x}_*, \tilde{\theta}\right)\right)$ and the vector $\zeta \sim \prod^{M} \mathcal{N}(0,1)$ are from M one-dimensional i.i.d. standard Gaussian distributions [129].

The mean predictions of the GP emulators for the 1000 Monte Carlo realisations are presented in Fig. 6.13 against the observational data used within BHM $z(\boldsymbol{x_z})$ with $\pm c_\sigma(V_{o,j} + V_{m,j})$ bounds; where c_σ is the standard deviation associated with 99% probability mass of a standard normal (assuming output distributions to be approximately Gaussian). Figure 6.13 demonstrates that all five outputs are within the defined uncertainty bounds. However large discrepancies between the experimental observations and simulator outputs (represented by the five emulator's mean predictions) occur, especially for the first and fifth natural frequencies. This illustrates that the simulator has model form errors, that would lead to incorrect parameter inference if model discrepancy was not considered in the calibration process.

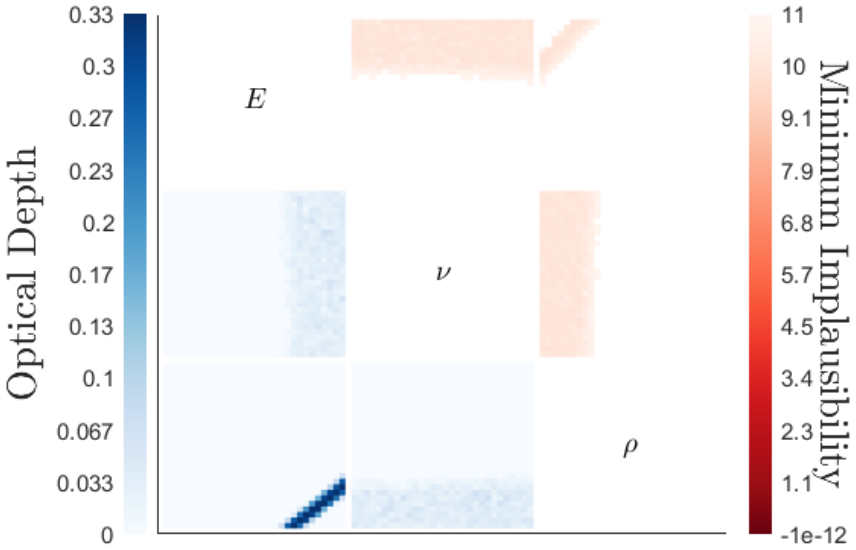


Figure 6.10: Minimum implausibility and optical depth plots for the first wave of BHM on the representatives five storey building structure. Each quadrant is a comparison of two parameter combinations for the given metric, e.g. the top right quadrant is ρ against E for minimum implausibility and the bottom left E against ρ for optical depth.

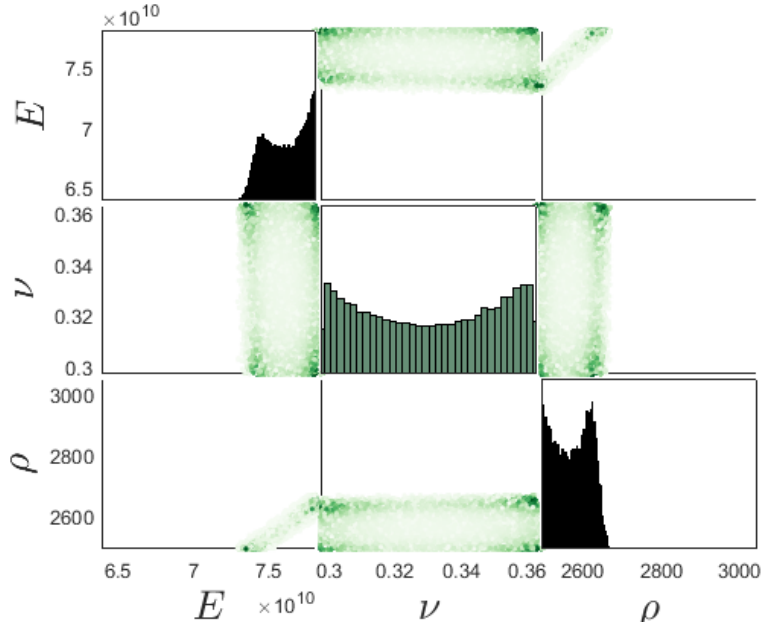


Figure 6.11: Marginal and pairwise joint posterior distributions for the first wave of BHM on the representatives five storey building structure, where a darker shade represents a higher probability.

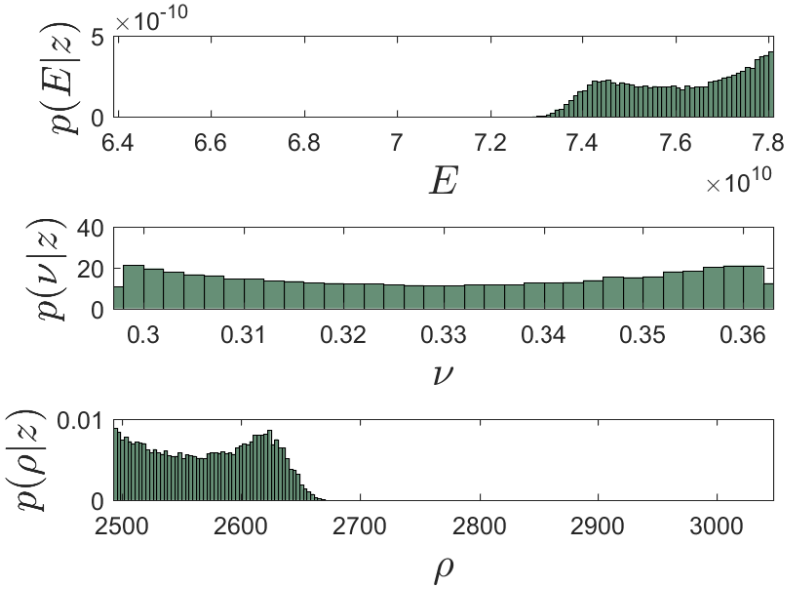


Figure 6.12: Marginal posterior distributions for the first wave of BHM on a representatives five storey building structure.

6.3.2 Model Discrepancy Learning via Importance Sampling

Assuming that the model discrepancy was additive and appropriately accounted for within the BHM process (and that the emulator mean accurately represents the simulator with negligible uncertainty), Fig. 6.13 shows valid samples from the approximate calibrated simulator outputs. The difference between these output samples and the observational data points give a visual indication of the model discrepancy magnitude and form. In order to infer the functional form of the model discrepancy Eq. (6.1) is redefined so that the model discrepancy remains additive but becomes functionally dependent on the inputs $\boldsymbol{x}$ and is assumed to be distributed as a GP i.e. $\delta(\boldsymbol{x}) \sim \mathcal{GP}\left(m(\boldsymbol{x}), K(\boldsymbol{x}, \boldsymbol{x}')\right)$ which is dependant on the hyperparameters $\boldsymbol{\phi}_\delta$.

The output predictive distribution from BHM displayed in Fig. 6.13 are samples $p\left(\boldsymbol{y}_{*,j}^{(i)} \mid \boldsymbol{x}_*, \boldsymbol{y}_j, \boldsymbol{x}_z, \boldsymbol{\theta}^{(i)}, \hat{\boldsymbol{\phi}}_{\eta,j}\right)$ with $\boldsymbol{\theta}^{(i)} \sim p\left(\boldsymbol{\theta} \mid Z, \boldsymbol{x}_z\right)$; where $\hat{\boldsymbol{\phi}}_{\eta,j}$ are the jth emulator's MLE estimate of the hyperparameters and Z is a matrix of the five outputs $z_j(\boldsymbol{x})|_{j=1:5}$. Assuming that $\hat{\boldsymbol{\phi}}_{\eta,j}$ for all j are appropriately estimated these can be assumed fixed rather than being marginalised out. This leads to an empirical Bayes assumption of the GP emulator hyperparameters for each output (as already assumed within BHM), and a consequence will be that uncertainty associated with $\boldsymbol{\phi}_{\eta,j}$ are not

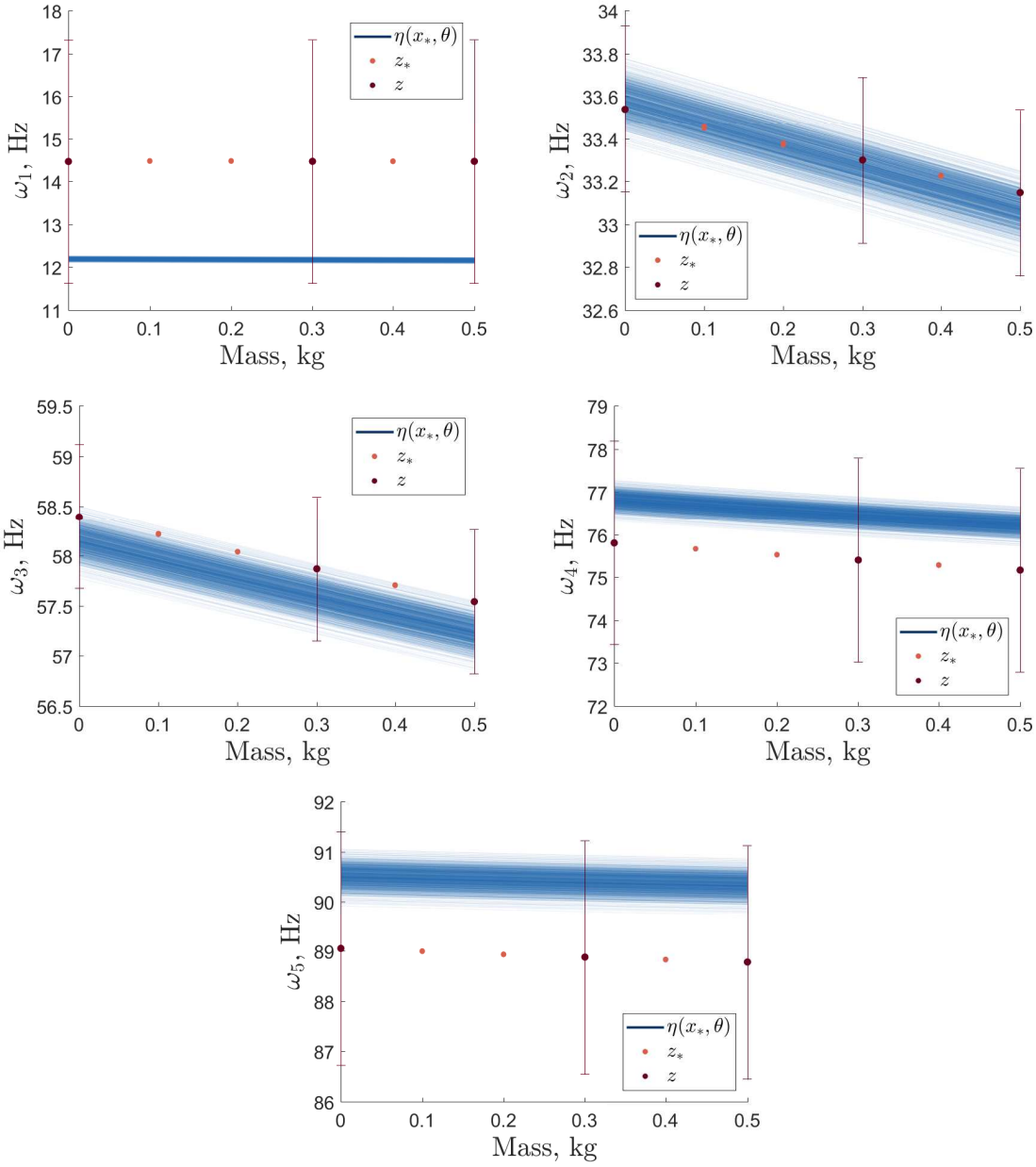


Figure 6.13: 1000 samples of the BHM predictive outputs, $p\left(\boldsymbol{y}_{*,j}^{(i)} \,|\, \boldsymbol{x}_*, \boldsymbol{y}_j, \boldsymbol{x}_z, \boldsymbol{\theta}^{(i)}, \hat{\boldsymbol{\phi}}_{\eta,j}\right)$ given $\boldsymbol{\theta}^{(i)} \sim p\left(\boldsymbol{\theta} \,|\, Z, \boldsymbol{x}_z\right)$.

incorporated; for this reason $\hat{\boldsymbol{\phi}}_{\eta,j}$ is removed in order to provide clarity of notation without loss of meaning.

The desired distribution for the jth output is $p\left(\boldsymbol{z}_{*,j} \,|\, \boldsymbol{x}_*, Z, \boldsymbol{y}_j, \boldsymbol{x}_z\right)$ where the model discrepancy hyperparameters $\boldsymbol{\phi}_{\delta,j}$ and parameters $\boldsymbol{\theta}$ are marginalised out. Calculating this distribution requires solving Eq. (6.30) where $p\left(\boldsymbol{z}_{*,j} \,|\, \boldsymbol{x}_*, Z, \boldsymbol{y}_j, \boldsymbol{x}_z, \boldsymbol{\theta}, \boldsymbol{\phi}_{\delta,j}\right)$ includes a GP mapping from $\boldsymbol{y}_j$ to $\boldsymbol{z}_j$ given $\boldsymbol{\theta}$, meaning that $\boldsymbol{y}_{*,j}$ is constructed from the GP emulator prediction for each output.

$$p\left(\boldsymbol{z}_{*,j} \,|\, \boldsymbol{x}_*, Z, \boldsymbol{y}_j, \boldsymbol{x}_z\right) = \int \left(\int p\left(\boldsymbol{z}_{*,j} \,|\, \boldsymbol{x}_*, Z, \boldsymbol{y}_j, \boldsymbol{x}_z, \boldsymbol{\theta}, \boldsymbol{\phi}_{\delta,j}\right) p\left(\boldsymbol{\phi}_{\delta,j}\right) \mathrm{d}\boldsymbol{\phi}_{\boldsymbol{\delta}} \right) \, p\left(\boldsymbol{\theta} \,|\, Z\right) \mathrm{d}\boldsymbol{\theta} \quad (6.30)$$

Equation (6.30) is intractable meaning that an approximation, using importance sampling can be formed in order to obtain samples from $p\left(\boldsymbol{z}_{*,j} \,|\, \boldsymbol{x}_*, Z, \boldsymbol{y}_j, \boldsymbol{x}_z\right)$. Section 6.3.2 presents three approaches for approximating Eq. (6.30) utilising importance sampling. These methods can also be seen as Bayesian model averaging, a technique outlined in Chapter 7 where models are averaged whilst weighted by their evidence.

Importance Sampling-Empirical Bayes

In the first approach, the inner integral (with respect to $\boldsymbol{\phi}_{\delta,j}$) is approximated using MLE estimates of the model discrepancy GP hyperparameters according to standard GP inference — this avoids calculating the inner integral. In contrast, the outer integral (with respect to $\boldsymbol{\theta}$) is approximated via importance sampling where the unnormalised proposal is given by samples from $p\left(\boldsymbol{\theta} \,|\, Z, \boldsymbol{x}_z\right)$, i.e. the re-sampled parameters from BHM — all of which are equally likely and therefore $q^{un}(\boldsymbol{\theta}^{(i)}) \propto 1$. For the ith parameter sample $\boldsymbol{\theta}^{(i)} \sim p\left(\boldsymbol{\theta} \,|\, Z, \boldsymbol{x}_z\right)$, outputs from the five independent GP emulators are obtained $\boldsymbol{y}_j^{(i)}|_{j=1:5}$. The outputs are used to train the ith model discrepancy GP in order to estimate $\hat{\boldsymbol{\phi}}_{\delta,j}^{(i)}$ and acquire the unnormalised weight $w^{un,(i)} = p\left(\boldsymbol{z}_j \,|\, \boldsymbol{y}_j^{(i)}, \boldsymbol{x}_z, \boldsymbol{\theta}^{(i)}\right)$ — which is the marginal likelihood of the model discrepancy GP; this is the case since the proposal is constant. By normalising these weights $p\left(\boldsymbol{z}_{*,j} \,|\, \boldsymbol{x}_*, Z, \boldsymbol{y}_j, \boldsymbol{x}_z\right)$ can be approximated as Eq. (6.31) were the mean and variance are obtained by the law of total expectation and variance Eqs. (6.32)

Algorithm 6 Importance Sampling-Empirical Bayes Model Discrepancy Inference

for $j = 1 : N_{out}$ **do**

Training;

for $i = 1 : N_s$ **do**

Predict $p\left(\boldsymbol{y}_j^{(i)} \,|\, \boldsymbol{y}_j, \boldsymbol{x}_z, \boldsymbol{\theta}^{(i)}, \hat{\boldsymbol{\phi}}_{\eta,j}\right)$

Optimise $\hat{\boldsymbol{\phi}}_{\delta,j}^{(i)} = \arg\max_{\boldsymbol{\phi}_{\delta,j}} p\left(\boldsymbol{z}_j \,|\, \boldsymbol{y}_j^{(i)}, \boldsymbol{x}_z, \boldsymbol{\theta}^{(i)}, \boldsymbol{\phi}_{\delta,j}\right)$

$w_j^{un,(i)} = p\left(\boldsymbol{z}_j \,|\, \boldsymbol{y}_j^{(i)}, \boldsymbol{x}_z, \boldsymbol{\theta}^{(i)}, \hat{\boldsymbol{\phi}}_{\delta,j}^{(i)}\right)$

end for

Normalise weights $w_j^{(i)} = w_j^{un,(i)} / \sum_{i=1}^{N_s} w_j^{un,(i)}$

Prediction;

for $i = 1 : N_s$ **do**

Predict $p\left(\boldsymbol{y}_{*,j}^{(i)} \,|\, \boldsymbol{x}_*, \boldsymbol{y}_j, \boldsymbol{x}_z, \boldsymbol{\theta}^{(i)}, \hat{\boldsymbol{\phi}}_{\eta,j}\right)$

Predict $\hat{\boldsymbol{z}}_{*,j}^{(i)}$ and $\Sigma_{z*,j}^{(i)}$ from $p\left(\boldsymbol{z}_{*,j}^{(i)} \,|\, \boldsymbol{x}_*, Z, \boldsymbol{y}_j, \boldsymbol{x}_z, \boldsymbol{\theta}^{(i)}, \hat{\boldsymbol{\phi}}_{\delta,j}^{(i)}\right)$

end for

Predict the approximation of $p\left(\boldsymbol{z}_{*,j} \,|\, \boldsymbol{x}_*, Z, \boldsymbol{y}_j, \boldsymbol{x}_z\right)$ via Eqs. (6.32) and (6.33)

end for

and (6.33).

$$p\left(\boldsymbol{z}_{*,j} \,|\, \boldsymbol{x}_*, Z, \boldsymbol{y}_j, \boldsymbol{x}_z\right) \approx \mathcal{N}\left(\mathbb{E}\left(\boldsymbol{z}_{*,j} \,|\, \boldsymbol{x}_*, Z, \boldsymbol{y}_j, \boldsymbol{x}_z\right), \mathbb{V}\left(\boldsymbol{z}_{*,j} \,|\, \boldsymbol{x}_*, Z, \boldsymbol{y}_j, \boldsymbol{x}_z\right)\right) \tag{6.31}$$

$$\mathbb{E}\left(\boldsymbol{z}_{*,j} \,|\, \boldsymbol{x}_*, Z, \boldsymbol{y}_j, \boldsymbol{x}_z\right) = \sum_{i=1}^{N_s} w_j^{(i)} \hat{\boldsymbol{z}}_{*,j}^{(i)} \tag{6.32}$$

$$\begin{aligned}\mathbb{V}\left(\boldsymbol{z}_{*,j} \,|\, \boldsymbol{x}_*, Z, \boldsymbol{y}_j, \boldsymbol{x}_z\right) = &\sum_{i=1}^{N_s} w_j^{(i)} (\Sigma_{z*,j}^{(i)} + \hat{\boldsymbol{z}}_{*,j}^{(i)} \hat{\boldsymbol{z}}_{*,j}^{(i)\mathsf{T}}) \\ &- \mathbb{E}\left(\boldsymbol{z}_{*,j} \,|\, \boldsymbol{x}_*, Z, \boldsymbol{y}_j, \boldsymbol{x}_z\right) \mathbb{E}\left(\boldsymbol{z}_{*,j} \,|\, \boldsymbol{x}_*, Z, \boldsymbol{y}_j, \boldsymbol{x}_z\right)^{\mathsf{T}}\end{aligned} \tag{6.33}$$

Where $\hat{\boldsymbol{z}}_{*,j}^{(i)}$ and $\Sigma_{z*,j}^{(i)}$ are the mean and variance of the jth GP mapping from $\boldsymbol{y}_j$ to $\boldsymbol{z}_j$ and N_s are the number of samples such that $\boldsymbol{\theta}^{(i)}|_{i=1:N_s}$. This process is summarised in Algorithm 6.

For this case study $N_s = 1000$ and the model discrepancy GPs are modelled with a zero mean and Matérn (where $p = 2$) plus Gaussian noise covariance functions (i.e. $K + \mathbb{I}\sigma_n^2$). Figure 6.14 states the output predictions of the five natural frequencies where model discrepancy has been accounted for. Apart from the first natural frequency where the training data does not adequately cover the input domain, the natural frequency predictions accurately account for the functional form of the model discrepancy. Validation of the results is discussed in Section 6.3.3.

Importance Sampling

The second approach approximates both integrals via importance sampling techniques. In this scenario the unnoramlised nominal distribution incorporates prior information about the hyperparameters, $p(\boldsymbol{\theta}) = p\left(\boldsymbol{z}_j \,|\, \boldsymbol{y}_j^{(i)}, \boldsymbol{x}_z, \boldsymbol{\theta}^{(i)}, \boldsymbol{\phi}_{\delta,j}^{(i,k)}\right) p\left(\boldsymbol{\phi}_{\delta,j}^{(i,k)}\right)$. Choices now remain as to the proposal distribution for $\boldsymbol{\phi}_{\delta,j}$. One option is to select the proposal to be equal to the prior i.e. $q^{un}\left(\boldsymbol{\phi}_{\delta,j}^{(i,k)}\right) = p\left(\boldsymbol{\phi}_{\delta,j}^{(i,k)}\right)$, this cancels with the prior in the nominal distribution meaning the unnormalised weights are $w_j^{un,(i,k)} = p\left(\boldsymbol{z}_j \,|\, \boldsymbol{y}_j^{(i)}, \boldsymbol{x}_z, \boldsymbol{\theta}^{(i)}, \boldsymbol{\phi}_{\delta,j}^{(i,k)}\right)$. The second choice is that the proposal is uniform $q^{un}\left(\boldsymbol{\phi}_{\delta,j}^{(i,k)}\right) \propto 1$ where the bounds are chosen to be large enough to have sufficient support over the target distribution — in this formulation the unnormalised weights are $w_j^{un,(i,k)} = p\left(\boldsymbol{z}_j \,|\, \boldsymbol{y}_j^{(i)}, \boldsymbol{x}_z, \boldsymbol{\theta}^{(i)}, \boldsymbol{\phi}_{\delta,j}^{(i,k)}\right) p\left(\boldsymbol{\phi}_{\delta,j}^{(i,k)}\right)$. Both these solutions should converge to the same approximation given enough samples. Algorithm 7 presents the process with which to marginalise out both $\boldsymbol{\phi}_{\delta,j}$ and $\boldsymbol{\theta}$. In this approach Eqs. (6.32) and (6.33) also become the sum over N_ϕ as $\hat{\boldsymbol{z}}_{*,j}^{(i,k)}$ and $\Sigma_{z*,j}^{(i,k)}$ are samples of both the hyperparameters and parameters, reflecting the discrete approximation of the double integral. Other importance sampling based approaches to marginalising the hyperparameters from a GP are adaptive importance sampling [165], where the proposal is iterative amended in order to improve convergence, and SMC [166]. These techniques could be implemented to provide faster convergence of the approximations.

Both choices of proposal have been implemented for this case study. Figure 6.15 presents the approximation when a uniform proposal is selected whereas Fig. 6.16 demonstrates the scenario when the proposal is equal to the prior. For both scenarios $N_\phi = 100$ and $N_s = 1000$ with the model discrepancy GPs modelled with a zero mean and Matérn (where $p = 2$) plus Gaussian noise covariance functions (i.e. $K + \mathbb{I}\sigma_n^2$). The log hyperparameter priors were Gaussian distributed — $\log \omega_{x,j} \sim \mathcal{N}(0, 6)$,

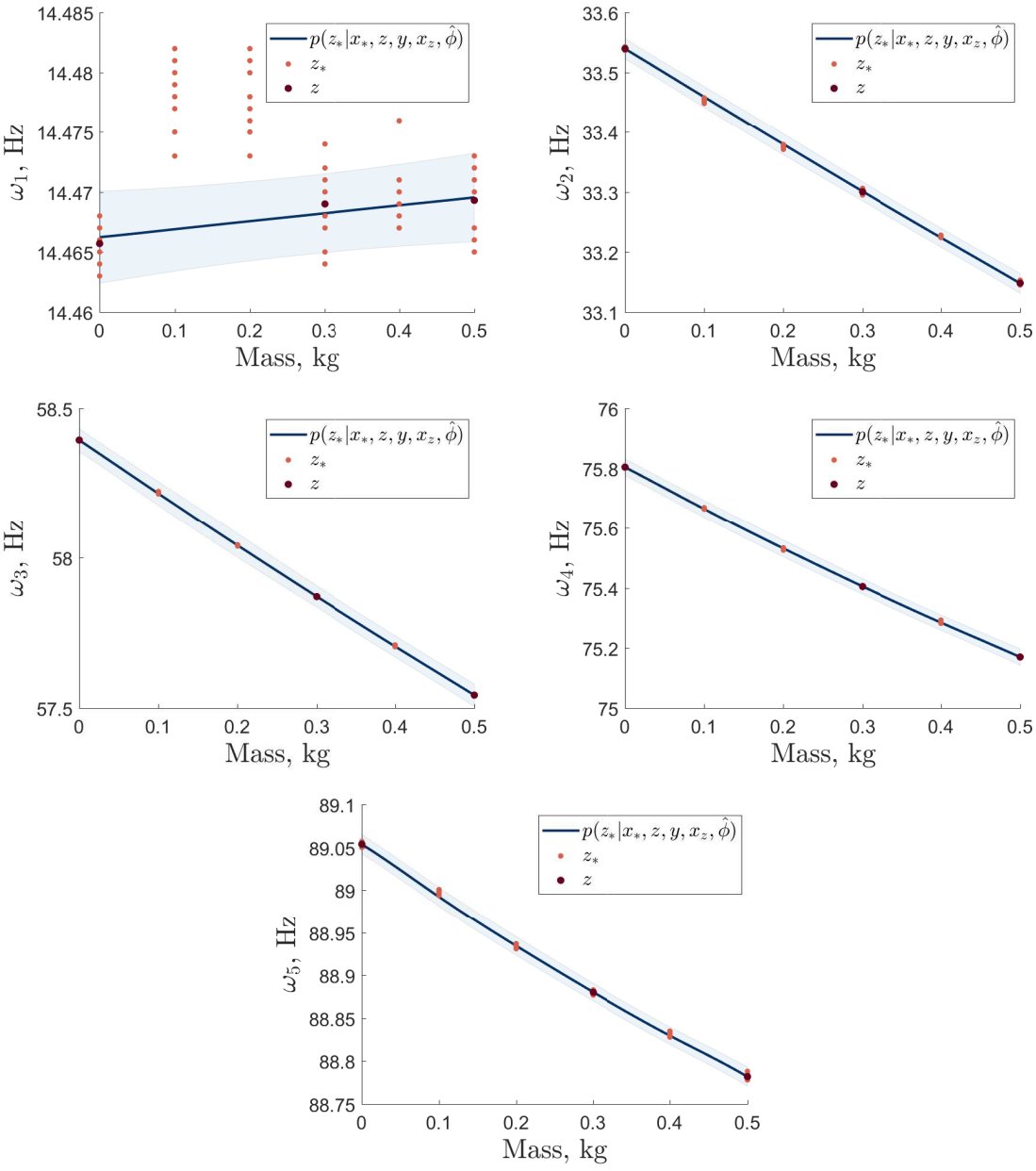


Figure 6.14: BHM predictive outputs with inference of model discrepancy via importance sampling and empirical Bayes trained GPs. The shaded regions indicate $\pm 3\sigma$.

Algorithm 7 Importance Sampling Model Discrepancy Inference

for $j = 1 : N_{out}$ **do**

Training;
for $i = 1 : N_s$ **do**
Predict $p\left(\boldsymbol{y}_j^{(i)} \,|\, \boldsymbol{y}_j, \boldsymbol{x}_z, \boldsymbol{\theta}^{(i)}, \hat{\boldsymbol{\phi}}_{\eta,j}\right)$
for $k = 1 : N_\phi$ **do**
Sample $\boldsymbol{\phi}_{\delta,j}^{(i,k)} \sim q\left(\boldsymbol{\phi}\right)_{\delta,j}$
$w_j^{un,(i,k)} = \frac{p\left(\boldsymbol{z}_j \,|\, \boldsymbol{y}_j^{(i)}, \boldsymbol{x}_z, \boldsymbol{\theta}^{(i)}, \boldsymbol{\phi}_{\delta,j}^{(i,k)}\right) p\left(\boldsymbol{\phi}_{\delta,j}^{(i,k)}\right)}{q^{un}\left(\boldsymbol{\phi}_{\delta,j}^{(i,k)}\right)}$
end for
end for
Normalise weights $w_j^{(i,k)} = w_j^{un,(i,k)} / \sum_{i=1}^{N_s} \sum_{k=1}^{N_\phi} w_j^{un,(i,k)}$

Prediction;
for $i = 1 : N_s$ **do**
Predict $p\left(\boldsymbol{y}_{*,j}^{(i)} \,|\, \boldsymbol{x}_*, \boldsymbol{y}_j, \boldsymbol{x}_z, \boldsymbol{\theta}^{(i)}, \hat{\boldsymbol{\phi}}_{\eta,j}\right)$
for $k = 1 : N_\phi$ **do**
Predict $\hat{\boldsymbol{z}}_{*,j}^{(i,k)}$ and $\Sigma_{z*,j}^{(i,k)}$ from $p\left(\boldsymbol{z}_{*,j}^{(i,k)} \,|\, \boldsymbol{x}_*, Z, \boldsymbol{y}_j, \boldsymbol{x}_z, \boldsymbol{\theta}^{(i)}, \boldsymbol{\phi}_{\delta,j}^{(i,k)}\right)$
end for
end for
Predict the approximation of $p\left(\boldsymbol{z}_{*,j} \,|\, \boldsymbol{x}_*, Z, \boldsymbol{y}_j, \boldsymbol{x}_z\right)$ via Eqs. (6.32) and (6.33)
end for

$\log \sigma_{fj}^2 \sim \mathcal{N}\left(\mathbb{V}\left(\boldsymbol{z}_j\right), 4\right)$ and $\log \sigma_{nj}^2 \sim \mathcal{N}\left(V_{o,j} - 5, 6\right)$ — stating that a low noise and smooth model discrepancy solution is expected. The bounds of the uniform proposal were $\{\log \omega_{x,j}, \log \sigma_{fj}^2, \log \sigma_{nj}^2\} \sim \mathcal{U}\left(\{-15, -25, -25\}, \{25, 10, 0\}\right)$ in order to provide adequate support over the hyperparameter domain.

In both Figs. 6.15 and 6.16 inclusion of the hyperparameter uncertainty inflates the predictive variance, given that the uncertainty associated with the hyperparameters is now approximated. The two choices of proposal both provide very similar predictions, which is expected if both are equally valid proposals and therefore approximations. The posterior distribution over the hyperparameters can also be estimated from the weights; Figs. 6.17 and 6.18 demonstrate these distributions for both proposal options for the fifth natural frequency. Both methods produce similar hyperparameter posteriors, however Fig. 6.17 clearly shows a much greater effect of the prior on the posterior distribution. This may mean that as the uniform proposal is generating samples over a wider, less focused hyperparameter domain the method may take longer to converge than using the prior as the proposal density. Furthermore both

results show the clear type I and II maximum likelihoods within the pairwise joint densities, where each refers to the noise and roughness invariant solutions.

Discussion

The aforementioned importance sampling based techniques for inferring model discrepancy require iteratively training a number of GP models, whether by importance sampling or by MLE estimates (i.e. an empirical Bayes approach). The computational complexity of these GP models is a function of the observational data length N_z, which for most applications will be very small. In this case study $N_z = 3$ making these approach computationally practical. However, in scenarios where the number of training observations is large the process could be run in parallel; where each emulator or each independent sample of the parameters could be run in parallel making the approach computationally practical.

The increased variance in Figs. 6.15 and 6.16 compared with Fig. 6.14 reflects the uncertainty in the hyperparameters and represent a more rigorous handling of the uncertainties. This can be most clearly seen in the first natural frequency where despite a lack of informative training data the solution has converged to constant mean process with a relatively large variance.

A benefit of inferring the functional form of the model discrepancy is that improvements to the simulator can be made. Figures 6.19 and 6.20 present examples of the observational predictions next to the model discrepancy for the fifth natural frequency. The model discrepancy functional form is easily extracted from the importance sampling techniques by removing the predictive emulator output samples $\boldsymbol{y}_{*,j}^{(i)}$ from $\hat{\boldsymbol{z}}_{*,j}^{(i)}$ before calculating Eqs. (6.32) and (6.33). These results show that the model discrepancy was relatively constant over the masses with a small linear slope for the fifth natural frequency. This information can then be used to improve the simulator, aiding the ability of BHM to appropriately approximate the parameter posterior $p\left(\boldsymbol{\theta} \,|\, Z, \boldsymbol{x}_z\right)$ with less variance due to $V_{m,j}$, which will in turn reduce the uncertainty in the predictive distributions. Lastly, multivariate GPs as both emulators and in modelling the model discrepancy may reduce the total uncertainty in the prediction. This would occur as if the outputs are assumed co-dependant on each other, then more information may be provided output about their functional form from their codependencies.

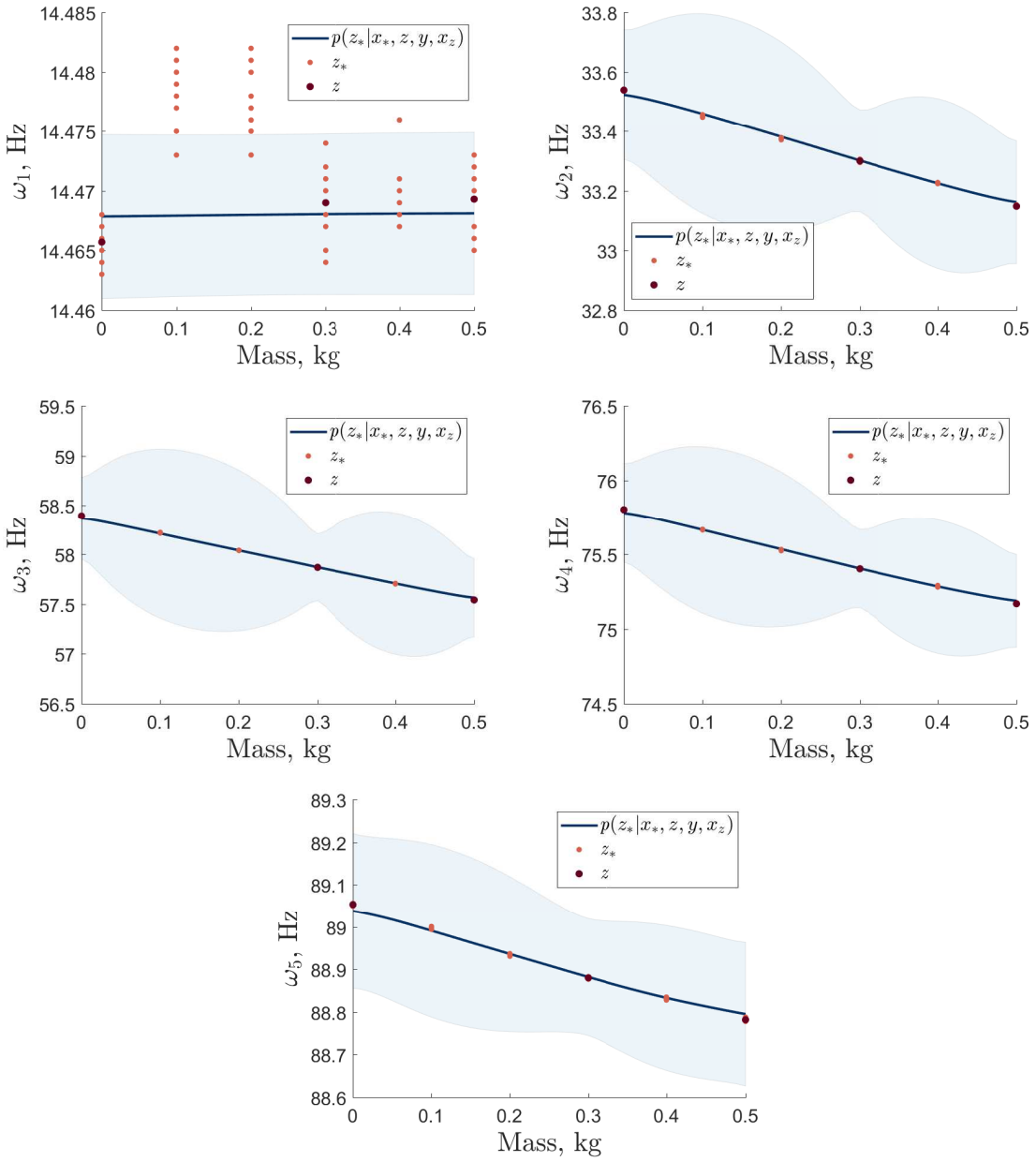


Figure 6.15: BHM predictive outputs with inference of model discrepancy and GP hyperparameters via importance sampling — uniform proposal. The shaded regions indicate $\pm 3\sigma$.

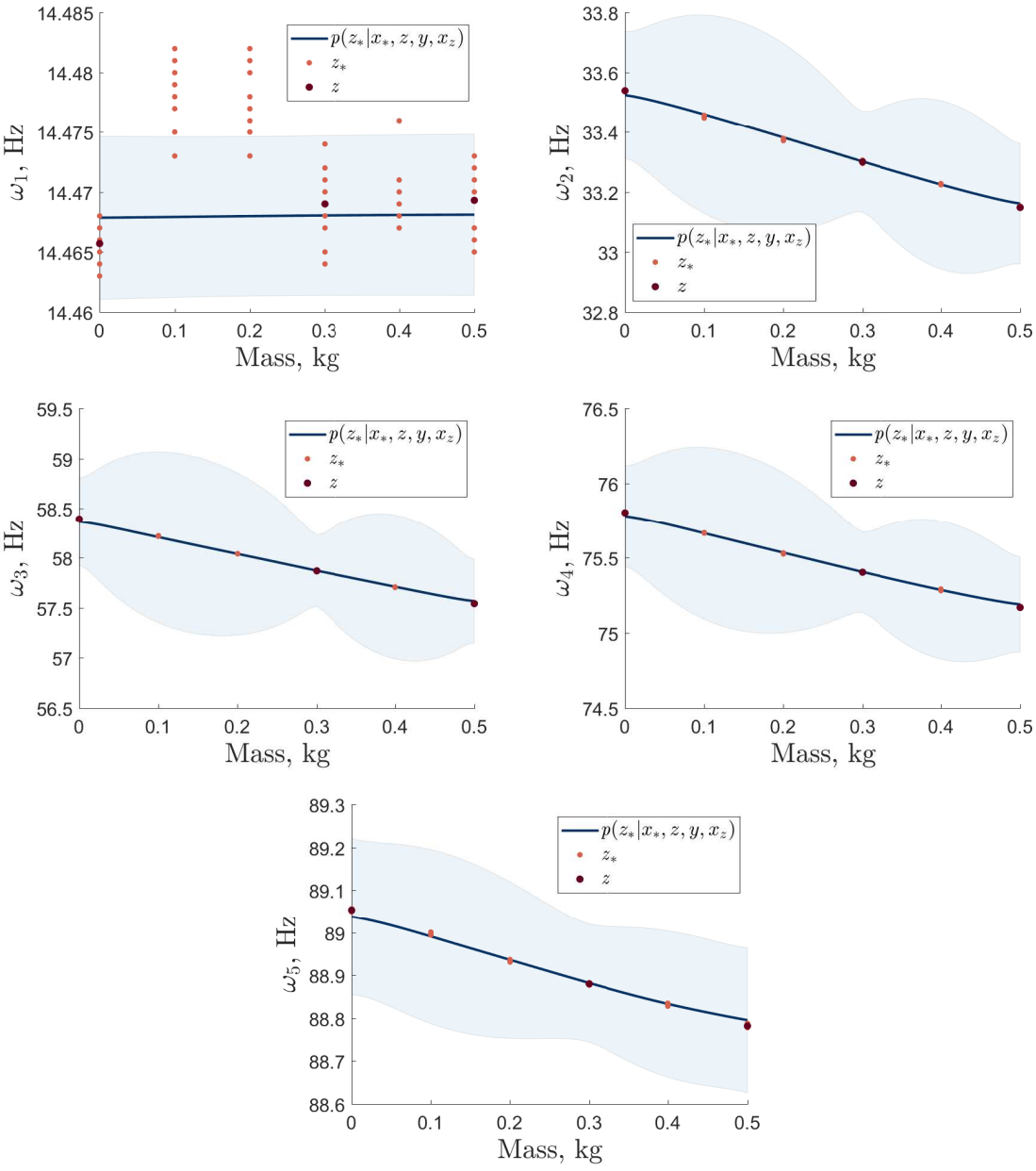


Figure 6.16: BHM predictive outputs with inference of model discrepancy and GP hyperparameters via importance sampling — prior proposal. The shaded regions indicate $\pm 3\sigma$.

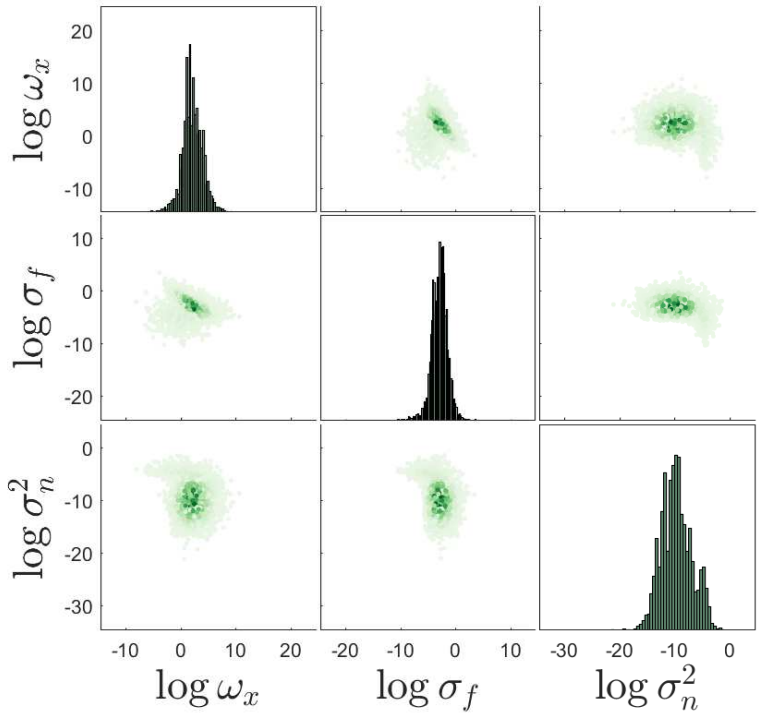


Figure 6.17: Marginal and pairwise joint posterior distributions of the hyperparameters given a uniform proposal for the fifth natural frequency.

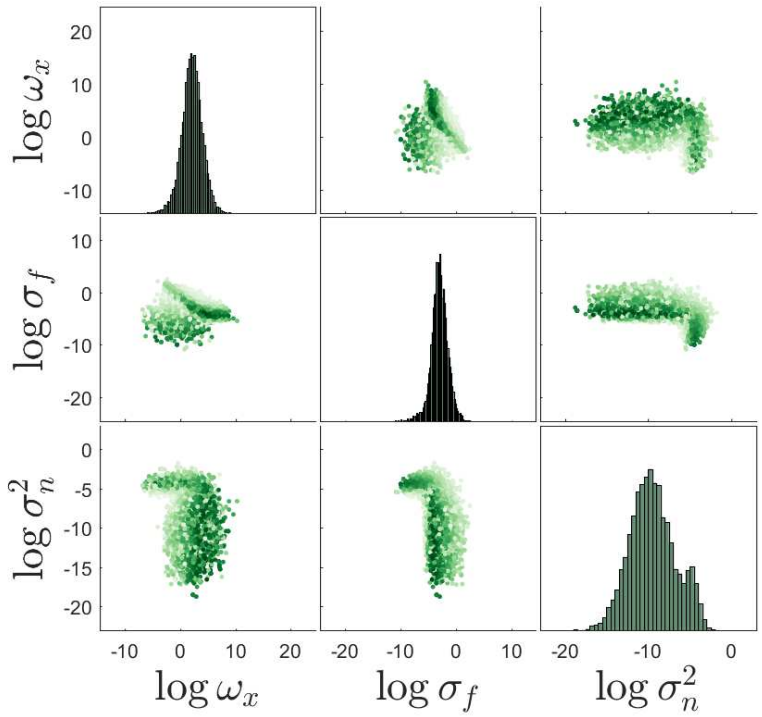


Figure 6.18: Marginal and pairwise joint posterior distributions of the hyperparameters given the proposal is the prior for the fifth natural frequency.

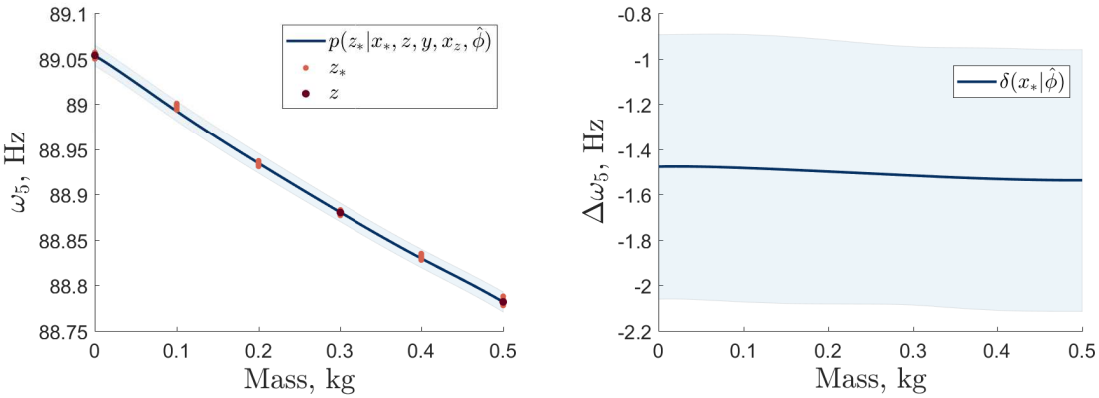


Figure 6.19: The predictions and model discrepancy for the importance sampling empirical Bayes approach for the fifth natural frequency. The shaded regions indicate $\pm 3\sigma$.

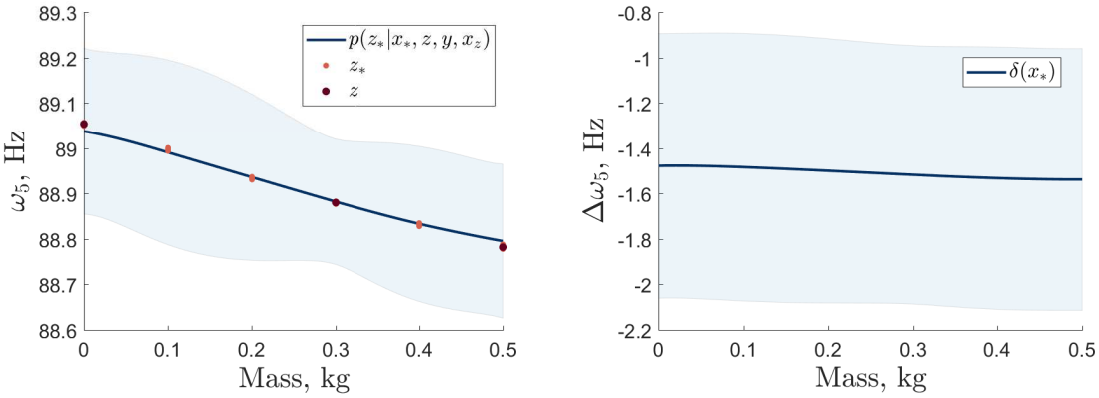


Figure 6.20: The predictions and model discrepancy for the importance sampling approach for the fifth natural frequency. The shaded regions indicate $\pm 3\sigma$.

6.3.3 Validation of Predictive Distributions

Hypothesis testing using both the KS- and MMD two sample tests were implemented with a significance level $\alpha = 5\%$ on the output predictions from both the importance sampling-empirical Bayes and importance sampling (with the proposal defined as the prior) approaches. Due to the low number of experimental data points, 100 repeats of ten samples were taken from the predictive distributions and averaged for the MMD-based hypothesis tests (implemented using a bootstrap approach with ten shuffles of the data set and a median heuristic to determine the hyperparameter of an SE kernel). The hypothesis test outcomes are presented in Tables 6.3 and 6.4 for the importance sampling-empirical Bayes approach and Tables 6.5 and 6.6 for the dual importance sampling technique.

The KS-tests indicate that for 50% of the predictions the null hypothesis could not be rejected for the importance sampling-empirical Bayes approach, compared to 6.6% for the importance sampling methodology. The MMD two sample tests confirm a similar interpretation, that for 33.3% of the predictions the null hypothesis could not be rejected compared to 13.3% for the importance sampling technique (where an averaged hypothesis ≥ 0.5 is consider rejected). These findings show that when the uncertainty associated with the hyperparameters is considered, the inflation of the variance leads to large dissimilarities between the observations and predictive distributions. By incorporating the uncertainty about the hyperparameters this will better reflect the lack of knowledge about the model discrepancy, but will also lead to increased confusion in classifications within an SHM approach, if decision bounds are based on these predictions. In addition, the importance sampling-empirical Bayes technique produces a worse prediction of the first natural frequency by not considering the uncertainty in the hyperparameters (as displayed in Fig. 6.14). The prediction is worse as it under-estimates the variance and the mean prediction fails to capture what is known physically, that natural frequency will decrease with added mass. The outcomes of the hypothesis test therefore show that the uncertainty within the modelling, observational data and model discrepancy are far to large to produce statistical representative predictions of the observational data.

To analyse the predictions further distance metrics were applied. Both the total variation and Hellinger distances, when predictions were compared to KDEs of the observational data (calculated via numerical integration), indicate that the importance sampling approach predictions are far from the observational data with most above 0.5 for the two metrics. According to these distances the first natural frequency distributions are close to the observational data for the 0kg, 0.3kg, 0.4kg and 0.5kg masses but far for the 0.1kg and 0.2kg cases. This is expected based on Fig. 6.15, where due to a lack of information about these states in the training data, the method fails to capture the majority of the observational points at these masses. The importance sampling-empirical Bayes technique in contrast shows much better performance, with the Hellinger distances below 0.5 for all but the 0.1kg and 0.2kg cases for the first natural frequency, which are further than the importance sampling distributions. In addition, the MMD distances confirm these trends, where for the second to fifth natural frequencies the importance sampling method predicts distributions far from the observational distributions, but predicts closer distributions for the 0.1kg and 0.2kg cases for the first natural frequency. The area metric for both approaches is relatively low, at an order of magnitude of 10^{-3}, caused by the close

Output	0.0kg	0.1kg	0.2kg	0.3kg	0.4kg	0.5kg
ω_1	0	1	1	1	0	0
ω_2	0	1	1	0	1	0
ω_3	1	1	1	1	1	1
ω_4	0	1	1	0	0	0
ω_5	0	1	0	0	0	0

Table 6.3: KS-test results for the importance sampling-empirical Bayes approach.

Output	0.0kg	0.1kg	0.2kg	0.3kg	0.4kg	0.5kg
ω_1	0.13	1.00	1.00	0.66	0.05	0.88
ω_2	0.42	0.69	0.71	0.02	0.97	0.43
ω_3	0.91	0.94	0.98	1.00	0.89	0.94
ω_4	0.65	0.87	0.81	0.88	0.66	0.80
ω_5	0.11	0.94	0.28	0.15	0.11	0.02

Table 6.4: MMD two sample test results for the importance sampling-empirical Bayes approach. Results are the average over 100 repeats of ten samples from the predictive distribution, using a bootstrap approach with ten shuffles and an SE kernel where the hyperparameters are determined by a median heuristic.

Output	0.0kg	0.1kg	0.2kg	0.3kg	0.4kg	0.5kg
ω_1	1	1	1	0	1	0
ω_2	1	1	1	1	1	1
ω_3	1	1	1	1	1	1
ω_4	1	1	1	1	1	1
ω_5	1	1	1	1	1	1

Table 6.5: KS-test results for the importance sampling approach.

Output	0.0kg	0.1kg	0.2kg	0.3kg	0.4kg	0.5kg
ω_1	0.40	1.00	1.00	0.12	0.29	0.34
ω_2	1.00	1.00	1.00	1.00	1.00	1.00
ω_3	1.00	1.00	1.00	1.00	1.00	1.00
ω_4	1.00	1.00	1.00	1.00	1.00	1.00
ω_5	1.00	1.00	1.00	1.00	1.00	0.99

Table 6.6: MMD two sample test results for the importance sampling approach. Results are the average over 100 repeats of ten samples from the predictive distribution, using a bootstrap approach with ten shuffles and an SE kernel where the hyperparameters are determined by a median heuristic.

Method	ω_1	ω_2	ω_3	ω_4	ω_5
Importance sampling-empirical Bayes	157.60	0.07	0.01	0.01	0.12
Importance sampling	145.11	0.51	0.25	0.34	0.94

Table 6.7: A comparison of NMSEs for the importance sampling-empirical Bayes and importance sampling approaches.

spacing of the observational points, leading to small areas between the empirical and predicted CDFs. The area metric also confirms the aforementioned differences in predictions.

NMSEs were quantified for the two approaches and displayed in Table 6.7. This deterministic view leads to the conclusion that the two approaches have accurately captured the mean trend for all natural frequencies apart from the first, with the importance sampling-empirical Bayes approach performing best for all but the first natural frequency.

These validation metrics all support the notion that these predicted outputs from both approaches would cause problems when utilised in a forward model-driven context, confusing decision bounds between damage states. However, the predictions from these approaches accurately reflect the uncertainties due to modelling, parameters, model discrepancy and observational noise. It is therefore challenging to produce predictions with reduced uncertainty without targeting each of these sources. The BHM-model discrepancy inference approaches described within this chapter provide a technique for understanding and targeting these uncertainty sources. By visualising and interrogating the model discrepancy functional form, simulator improvements and model selection can be targeted in a more rigorous manner. Furthermore, in scenarios where the training data is representative of the other known states, MLE estimates of the hyperparameters can be appropriate. In contrast, using MLE estimates of the hyperparameters in scenarios, like the first natural frequency, where the training data is not representative of the remaining states, will lead to overly confident uncertainty estimation when compared to marginalising the hyperparameters out. Furthermore, issues will always arise when the number of observational points and repeats are low. Validation metrics will struggle to accurately reflect the differences when they are constructed from a small number of observational samples, as in this case study.

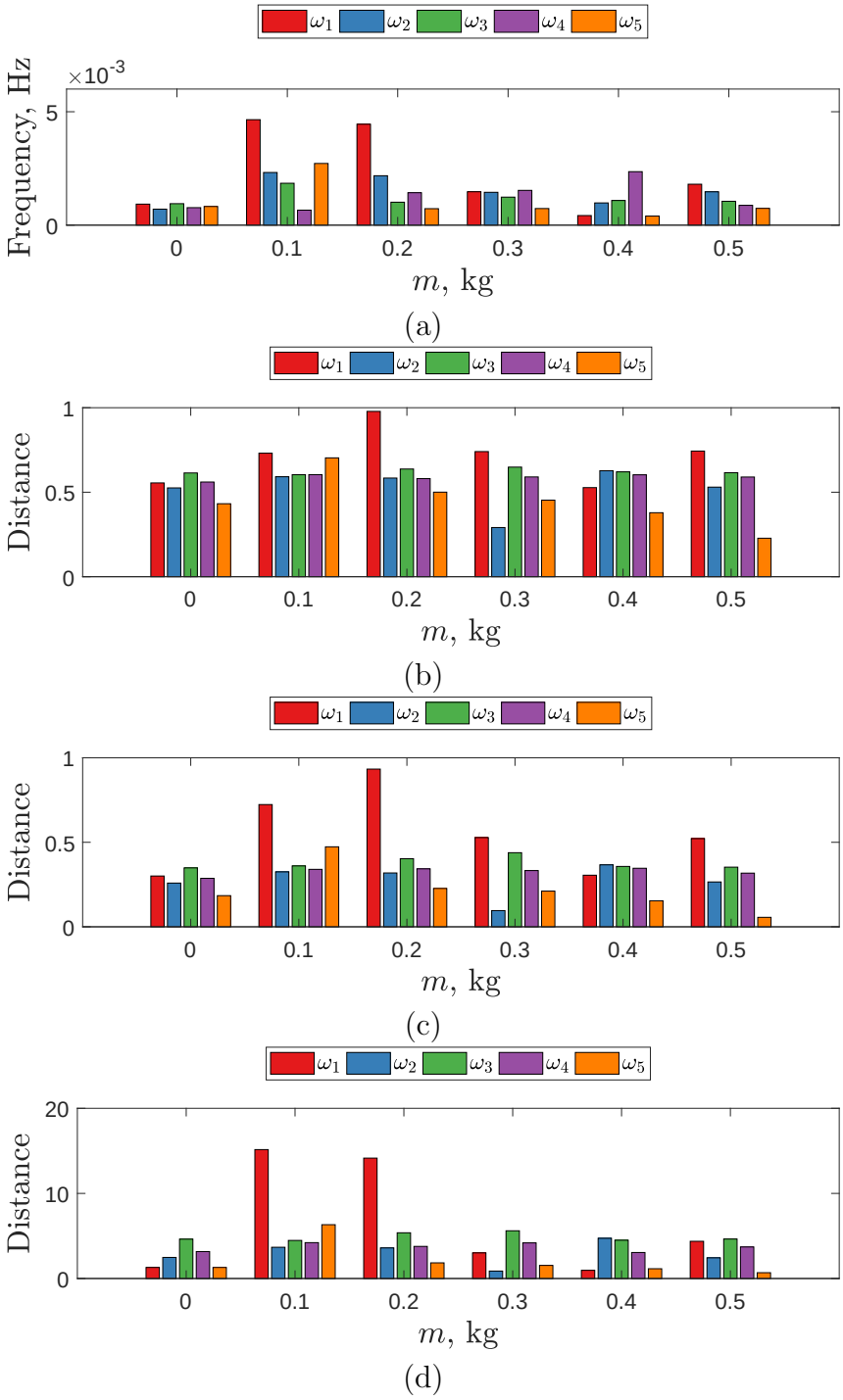


Figure 6.21: Statistical distances applied to the predictions from the importance sampling-empirical Bayes approach. Panel (a) is the area metric when compared to the empirical ten point observational CDF. Panel (b) and (c) are the total variation and Hellinger distances when compared to KDEs of the observational data. These three distance metrics have been calculated via numerical integration. Panel (d) is the averaged MMD distance over 100 repeats of ten samples from the predictive distribution.

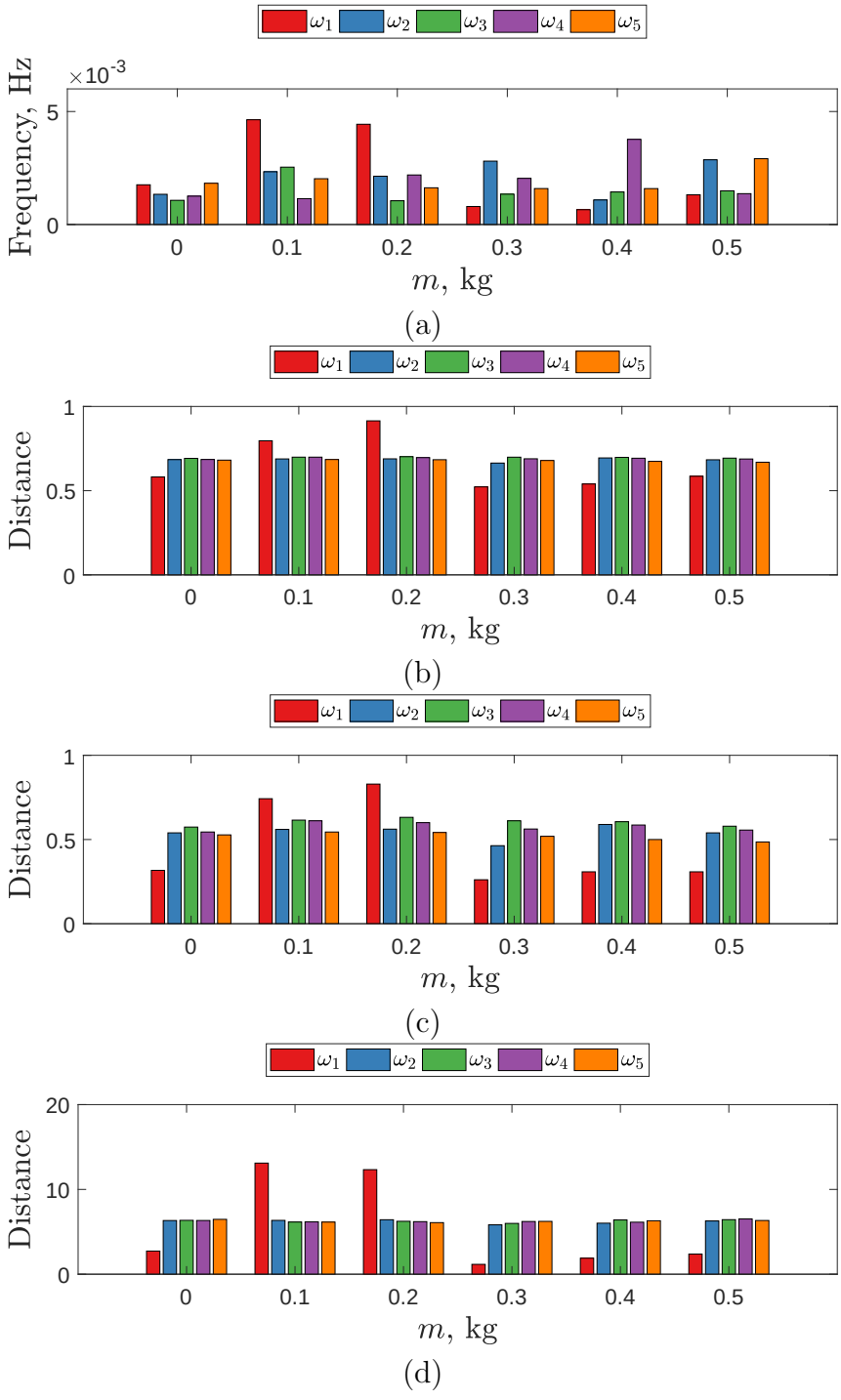


Figure 6.22: Statistical distances applied to the predictions from the importance sampling. Panel (a) is the area metric when compared to the empirical ten point observational CDF. Panel (b) and (c) are the total variation and Hellinger distances when compared to KDEs of the observational data. These three distance metrics have been calculated via numerical integration. Panel (d) is the averaged MMD distance over 100 repeats of ten samples from the predictive distribution.

6.4 Conclusion

BHM is an effective method for discarding parameter space in an iterative and 'likelihood free' manner. This approach means that difficult to emulate outputs or input combinations can be excluded and reintroduced between waves when they are more defined; which would not be possible in a likelihood based approach. Additionally, the method can be considered a specific case of ABC which has been shown to perform exact Monte Carlo inference given additive uniform model discrepancy. This removes the non-identifiablity problems of BCBC by separating out the model discrepancy inference with the parameter distribution. It has also been shown that the posterior parameter distributions can be approximated by importance sampling.

Sequential approaches to BHM provide a more efficient approach to designing the locations of simulator evaluations. Heuristics such as probability of non-implausibility, and expected (un)improvement provide one view of integrating sequential designs into BHM. Alternatively information-based techniques should provide a more efficient process for selecting simulator evaluations, although these are left as areas of further research. Moreover, more informative methods of sampling the parameter domain within BHM should be explored, such as an SMC-BHM approach.

The representative five storey building structure case study demonstrated the ability of BHM to calibrate and identify posterior parameter distributions given a simulator with pronounced model form errors. In addition to approximating the posterior parameter distribution via importance sampling, a methodology was presented for identifying the functional form and uncertainty associated with model discrepancy via marginalising out the parameters, and a method for marginalising out the hyperparameters of the model discrepancy GP. The predictions from these approaches captured the uncertainties associated with model discrepancy, with mean predictions that accurately capture the behaviour of the natural frequencies. However the resulting increased variance meant that these predictive distributions were statistically significantly dissimilar to those from the observation samples. This shows a problem with obtaining only a small number of observational samples, as well as the challenges introduced by a more rigorous handling of uncertainty.

Chapter 7

Multi-Level Uncertainty Integration

The main objective of forward model-driven SHM is to solve problems associated with the lack of available damage state data at a full-system level. As a consequence forward model-driven methods must employ a strategy that produces confidence in full-system predictions of health states without a traditional approach to validation at the full-system level. This provides the motivation for developing a multi-level uncertainty integration strategy.

Multi-level uncertainty integration is a process whereby a structure is divided into levels, where at the top is the full-system and below are potentially multiple levels of sub-systems, each with a number of simulators. For each sub-system it is expected that damage state data can be obtained, therefore allowing the simulators at this level to be calibrated and validated. The approach contains a mechanism for these validated sub-system simulators to be incorporated in a full-system level, providing a level of validation and confidence. The key assumption is that UQ can be adequately performed at multiple sub-system levels and that all damage mechanisms of interest can be understood at a sub-system level.

In Section 7.2 a subfunction discrepancy approach is outlined. The technique seeks to capture model discrepancies for each sub-system simulator and subsequently validate the bias corrected predictions. These discrepancies are propagated through to a full-system level with each simulator's parameter uncertainties providing improved

confidence in the full-system predictions and correcting simulator inadequacies. The following chapter presents a subfunction discrepancy approach within a multi-level uncertainty integration strategy, before demonstrating the technique on a numerical example and outlining conclusions.

7.1 Introduction

Improving complex system predictions by incorporating knowledge from multiple simulators (or modelling sources) has been attempted through a variety of techniques all with differing objectives. Examples of approaches are Bayesian model averaging, multi-fidelity and multi-level UQ. These methods are introduced and discussed, relating their applicability in resolving a key challenge in forward model-driven SHM, namely the problem of a lack of available data at a full-system level.

Bayesian model averaging removes the concept of a 'best' simulator and instead considers an ensemble of plausible simulators that are assumed to be from a distribution; where all the simulators attempt to model the same phenomena. By considering the weighted predictions from the ensemble a more reliable forecast can be made. By defining an ensemble of N simulators $\{\mathcal{M}_1, \ldots, \mathcal{M}_N\}$ trained using a set of observations $\mathcal{D}$ in order to predict $\boldsymbol{y}$, a set of posteriors, i.e. $p(\boldsymbol{y} \mid \mathcal{M}_i, \mathcal{D})$ for the ith simulator, can be obtained through a variety of Bayesian techniques. The law of total probability means that the posterior $p(\boldsymbol{y} \mid \mathcal{D})$ when the ensemble of simulators is marginalised out becomes the sum in Eq. (7.1).

$$p(\boldsymbol{y} \mid \mathcal{D}) = \sum_{i=1}^{N} p(\mathcal{M}_i \mid \mathcal{D}) p(\boldsymbol{y} \mid \mathcal{M}_i, \mathcal{D}) \tag{7.1}$$

Essentially Bayesian model averaging involves weighting posterior predictions from each simulator by the likelihood of the particular simulator being correct given training data, i.e. $w_i = p(\mathcal{M}_i \mid \mathcal{D})$, where w_i denotes a weight. The technique has found multiple uses in improving forecasts/extrapolations [167–170]. The method is well suited to weather forecasts [167, 169], where the system of interest is complex, time-varying data is available, model bias can be present and in different amounts varying with application context. In contrast, forward model-driven SHM applications will often not have data, at least initially, for any damage extents at a full-system

level, making the technique not viable as a solution to the lack of full-system data problem. On the other hand, the approach could be used to improve sub-system level predictions where damage state data is available.

Multi-fidelity modelling is similar to Bayesian model averaging but covers the more general case of using multiple approximate simulators to inform and accelerate UQ. The objective is that by using several low-fidelity simulators (or even in combination with a high-fidelity simulator) increased speed (typically when using Monte Carlo simulations) and accuracy can be achieved. The approach is more than just emulation as it often links low-fidelity models with a high fidelity model in order to estimate accuracy and provide convergence guarantees on the outputs from UQ methodologies. A review of these approaches is provided by Peherstorfer et al. in [171]. As with Bayesian model averaging these techniques could provide savings within UQ applications where one system is being modelled by several representations and data is available for this system. The methodologies do not provide a clear solution to a scenario where full-system data is not obtainable, like in forward model-driven SHM.

Multi-level uncertainty integration (or multi-level modelling) on the other hand provides a strategy for combining simulators at different levels in order to make full-system predictions. This can be from a combination of deterministic and/or stochastic simulators [172]. The approach is often formulated as Bayesian inference where there is some unknown set of global parameters Θ that are common to a set of simulators; which are either nominally similar with slight modifications in parameters [172] or have variations in boundary conditions and loading [55]. The Bayesian formulation allows the construction of a graphical model in order to visualise the conditional relationships between variables and simulator interactions, with inference performed using an MCMC scheme for the complete variable set. A Bayesian network/graphical model states the conditional probability relationships where an arrow denotes conditionality (and therefore is a directed graph). For example in Fig. 7.1, starting at the node for $\boldsymbol{\theta}$ there is only one arrow connected to the node $\boldsymbol{\eta}$; this can be written mathematically as $p\left(\boldsymbol{\eta} \,|\, \boldsymbol{\theta}\right)$.

Sankararaman and Mahadevan attempt to formalise this approach by providing a distinction between two types of simulator interaction; type-I, where the simulator outputs are inputs to the next simulator illustrated in Fig. 7.1, and type-II where simulator outputs can be determined from parameters inferred from another simulator as presented in Fig. 7.2 [55]. However, multiple other interactions could be imagined, for example a combination of type-I and -II together, or a scenario where discrepancy

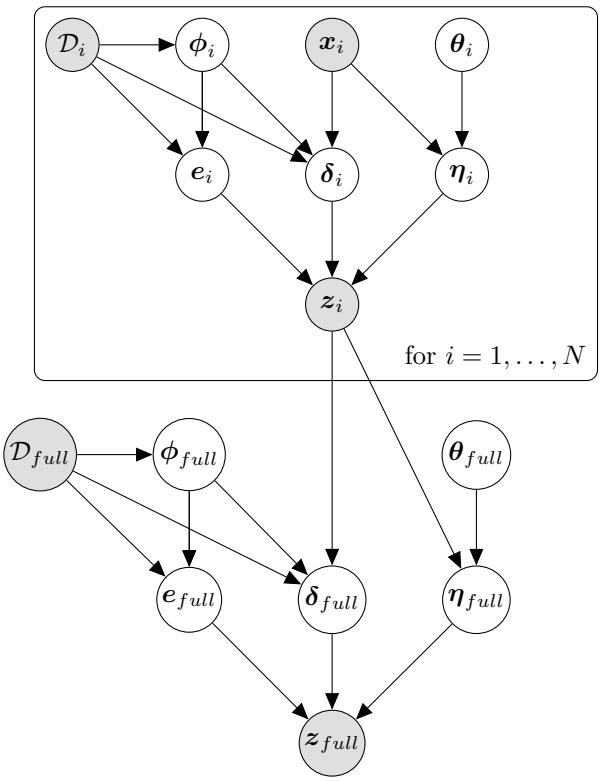


Figure 7.1: Bayesian network for a type-I interaction.

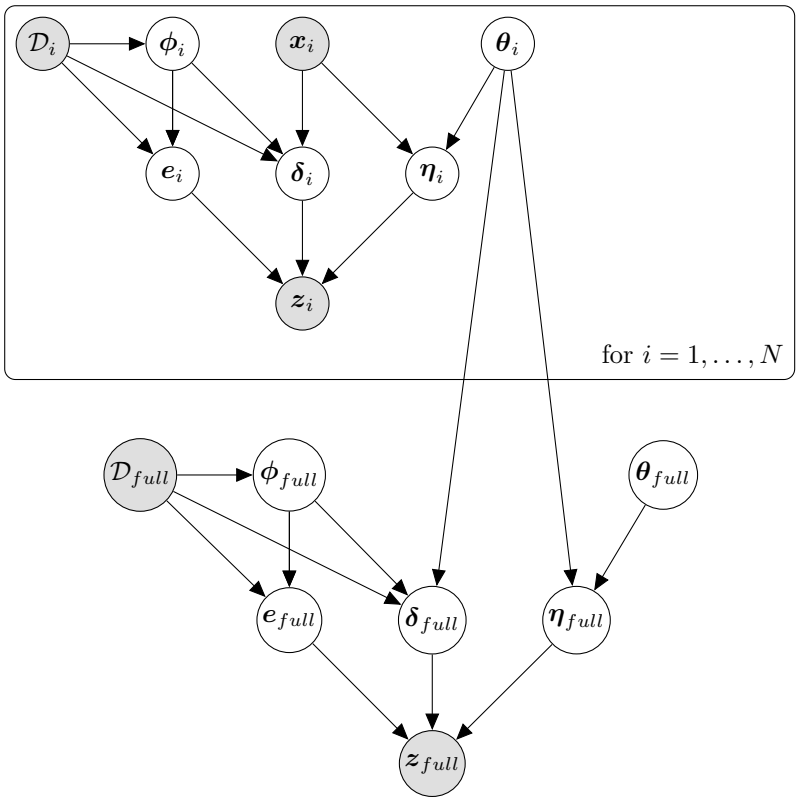


Figure 7.2: Bayesian network for a type-II interaction.

is excluded from the output conditional, or where discrepancy is the only quantity that the following simulator is conditioned on. These divisions may not be completely general and therefore not be appropriate in a complex system of sub-simulators.

Additionally, inference of a large combination of simulators, all with different data sets, will become increasingly costly and difficult to perform in one step. Instead it may be more practical to combine simulators that have been calibrated separately and perform full-system inference by Monte Carlo sampling each simulator (or its equivalent emulator). Furthermore, these frameworks often fail to discuss the complexities of simulator interactions when modelling components or sub-assemblies of a larger structure. Although general graphical models like those in Figs. 7.1 and 7.2 can be created for any arbitrary simulator, practical difficulties often arise when dividing a structure into approach sub-systems and levels.

Another view of multi-level uncertainty integration, similar to the general Bayesian inference methods, is that of the subfunction discrepancy approach. The technique was initially developed as a method for combining medical trials and health models together to make inferences about whether to fund drug trials [173]. The approach seeks to divide a full-system into a series of known or measurable sub-systems at differing levels. These are then combined (originally as type-I interactions) in order to make decisions about the cost of funding a certain treatment. Data about the costs associated with funding or not funding the treatment for the complete population are not obtainable and therefore each sub-system simulator must be accurate, based on observed data at that level. This means that model discrepancy is inferred at each state and propagated through. This technique allows calibration to be performed on a simulator by simulator basis. This type of approach is well suited to forward model-driven SHM as it provides a framework for obtaining confidence in full-system predictions without requiring data at the full-system level. The technique could also be formulated as a graphical model, however this is not investigated in this chapter.

7.2 A Subfunction Discrepancy Approach

Forward model-driven SHM relies on the ability to generate validated simulators of full-systems for various damage mechanisms of interest. A complication is that observational data for each of these damage states are often not obtainable at a full-system level. This creates a problem in how to validate and gain confidence in a

full-system simulator especially when modelling damage scenarios where observational data is not available. This provides the key motivation behind the development of a multi-level uncertainty integration strategy for forward model-driven SHM. This type of strategy seeks to use a combination of validated sub-system simulators at various levels, e.g. at material, component or sub-assembly level, in order to capture behaviours of the full-system, for which observational data cannot be obtained, with the aim of producing the required outputs under these behaviours. In an SHM context, the desired full-system outputs are damage sensitive features and the behaviours are generally the changes of these features under damage types of interest (although additional environmental changes may be included).

A strategy for performing multi-level uncertainty integration is a subfunction discrepancy technique. Here the approach is an adaptation and unification the approach proposed by Strong et al. in [173] for forward model-driven SHM. The proposed method divides a full-system structure into a number sub-system simulators that meet certain requirements. Firstly, there must be an output that can be measured experimentally and used to validate the sub-system simulator for the required inputs $\boldsymbol{x}^{sub} \in \boldsymbol{x}^{full}$ (where the superscripts sub and $full$ indicate the sub-system and full-system levels). Secondly, it is imperative that the functional relationship contains a set of parameters $\boldsymbol{\theta}^{sub} \in \boldsymbol{\theta}^{full}$ and inputs $\boldsymbol{x}^{sub} \in \boldsymbol{x}^{full}$ that are included in or affect the full-system simulator. Accordingly, experimental data at each sub-system can be used to validate and make inferences about their respective simulator. UQ for each sub-system may be employed to quantify parameter, model discrepancy and observational uncertainties, leading to confidence in the sub-system simulator. These inferences can be propagated through to the next level of the strategy. A key assumption is that the physics controlling changes at a full-system level can be captured at multiple sub-system levels, in addition to corrections for model discrepancies and quantification of uncertainties. Once propagated to a full-system level, this results in a complete understanding of the full-system uncertainties and should reduce model form errors or missing physics to a negligible level for the desired output quantity.

The methodology assumes that simulators at a sub-system level, for which observational data $\boldsymbol{z}$ is obtainable, can be modelled statistically in Eq. (7.2).

$$z(\boldsymbol{x}) = y(\boldsymbol{x}) + e = \eta(\boldsymbol{x}, \boldsymbol{\theta}) + \delta(\boldsymbol{x}) + e \tag{7.2}$$

Where $z(\boldsymbol{x})$ and $y(\boldsymbol{x})$ are observational and bias corrected simulator outputs given

the inputs $\boldsymbol{x}$ respectively. The bias corrected simulator output is equal to the sum of the simulator $\eta(\boldsymbol{x}, \boldsymbol{\theta})$ and the model discrepancy $\delta(\boldsymbol{x})$, where $\boldsymbol{\theta}$ are parameters of the simulator. The observations are assumed to be uncertain reflected in the addition of e.

The subfunction method for the simplest case — one level, one sub-system simulator and one full-system simulator, where all the parameters are contained within the sub-system simulator — can be defined mathematically as in Eqs. (7.3) and (7.4).

$$y^{sub}(\boldsymbol{x}) = \eta^{sub}(\boldsymbol{x}, \boldsymbol{\theta}) + \delta^{sub}(\boldsymbol{x}) \tag{7.3}$$

$$z^{full} = y^{full}(\boldsymbol{x}) + e^{full} = \eta^{full}(y^{sub}(\boldsymbol{x}), \boldsymbol{\theta}) + e^{full} \tag{7.4}$$

In this case, by performing inferences and validating the sub-system level simulator using Eq. (7.3), confidence can be established for the outputs of the full-system model under the input $\boldsymbol{x}$. It is assumed in this example that model discrepancy is found only at the sub-system level. This means that the model discrepancy at a full-system is only dependent on that of the sub-system. It is noted that it is possible to add model discrepancy at a full-system level (Eq. (7.4)); this maybe useful for correcting non-input dependant model form errors. In Eq. (7.2) the model discrepancy is described as having a functional form, implying that the simulator does not include all physics and may have assumptions or approximations that affect the functional output. As stated in Section 2.1.3, considering the functional form of model discrepancy is important for making robust statistical inferences. As a result, the subfunction discrepancy approach proposed utilises GP models in order to infer model discrepancy. Due to the bespoke nature of the subfunction discrepancy approach Section 7.3 provides an explanation of the technique applied to a numerical case study of a four degree-of-freedom shear structure.

7.3 Shear Structure Case Study

A numerical case study is presented as a demonstration of the subfunction discrepancy approach to multi-level uncertainty integration outlined in Section 7.2. The full-system is a linear four degree-of-freedom shear structure where each of the four masses

is supported by a bolted beam illustrated Fig. 7.3a. The objective is to calculate the distributions of the four natural frequencies $\boldsymbol{\omega}_n = \{\omega_1, \omega_2, \omega_3, \omega_4\}$, as two damage types are introduced to the structure — an open crack of length l_{cr} at the midpoint of any of the four beams, and a reduction in the non-dimensionalised clamping force f, at any of the four bolted joints. For simplicity both damage types are assumed to only affect the stiffness K of the full-system and in a quasi-static manner where $K = (K_b \times K_j)/(K_b + K_j)$ (where K_b and K_j are the beam and joint stiffnesses respectively). Accordingly, the functional mapping of the full-system is defined as $X^{full} \rightarrow \boldsymbol{y}^{full}$ where $\boldsymbol{y}^{full} = \boldsymbol{\omega}_n$, $X^{full} = \{\boldsymbol{l_{cr}}, \boldsymbol{f}\}$ and the mapping depends on parameters $\boldsymbol{\theta}^{full}$. In addition, parameter distributions in this case study are fixed for both simulator and 'true' behaviours, stated in Table 7.1. Calibration is not pursued in this case study in order to simplfy the explanation of the framework, although it is possible to implement calibration techniques (such as BCBC or BHM) within the approach. The parameter distributions have been chosen so that a wide range of distributions are represented in order to display the flexibility of the technique. Throughout this case study UP is performed via Monte-Carlo sampling using 500 draws from the parameter distributions in simulated cases. For experimental tests 50 repeats are performed, where each repeat is an independent draw for the parameter distributions.

The first stage of the subfunction discrepancy approach is to divide the full-system into corresponding sub-systems, for which there must be measurable outputs and either parameters or outputs that affect the full-system. This study divides the four degree-of-freedom shear structure by one level at which there are two sub-systems — the beam and bolted joint — presented in Fig. 7.3. The reason for this is that static deflection tests can be performed for an increase in damage in each sub-system, and subsequently, quasi-static stiffness values can be determined from the experimental tests. Additionally, both sub-systems inform of the full-system response as each can be used to quantify the stiffness reduction under their respective damage type. The sub-systems can be defined as follows. For the beam sub-system (simulator one, level one): $\eta^{sub}_{1,1} : \boldsymbol{x}^{sub}_{1,1} \rightarrow \boldsymbol{y}^{sub}_{1,1}$ where $\boldsymbol{x}^{sub}_{1,1} = \boldsymbol{l}_{cr}$, and $\boldsymbol{y}^{sub}_{1,1} = \boldsymbol{K}_b$ the beam tip stiffness. For the joint sub-system (simulator two, level one): $\eta^{sub}_{2,1} : \boldsymbol{x}^{sub}_{2,1} \rightarrow \boldsymbol{y}^{sub}_{2,1}$ where $\boldsymbol{x}^{sub}_{2,1} = \boldsymbol{f}$, and $\boldsymbol{y}^{sub}_{2,1} = \boldsymbol{K}_j$ the joint stiffness. A full-system simulator is then formed as $\eta^{full} : \boldsymbol{x}^{full} \rightarrow \boldsymbol{y}^{full}$ for the inputs $X^{full} = \{\boldsymbol{n}_{cr}, \boldsymbol{n}_f, y^{sub}_{1,1}(\boldsymbol{x}^{sub}_{1,1}), y^{sub}_{2,1}(\boldsymbol{x}^{sub}_{2,1})\}$ and outputs $\boldsymbol{y}^{full} = \boldsymbol{\omega}_n$. The following subsections describe how each system is constructed with details on the simulator and numerical experimental data.

Parameter		**Value**	**System**	**'True'**	**Simulator**
Beam length	l_b	$175mm$	$sub_{1,1}$, $full$	✓	✓
Beam width	w_b	$25mm$	$sub_{1,1}$, $full$	✓	✓
Beam thickness	t_b	$5mm$	$sub_{1,1}$, $full$	✓	✓
Plate length	l_p	$300mm$	$full$	✓	✓
Plate width	w_p	$250mm$	$full$	✓	✓
Plate thickness	t_p	$25mm$	$full$	✓	✓
Elastic Modulus	E	$\mathcal{N}(71, 0.5^2)GPa$	$sub_{1,1}$, $full$	✓	✓
Density	ρ	$\mathcal{N}(2700, 100^2)$ kg/m^3	$full$	✓	✓
Beam length crack location	x_{cr}	$87.5mm$	$sub_{1,1}$	✓	✓
Crack model parameter	α	0.667	$sub_{1,1}$	✓	
Initial joint stiffness	K_{ji}	$\mathcal{W}ei(50005, 100)$ N/m	$sub_{1,2}$, $full$	✓	✓
Rate of joint stiffness change	κ	$\mathcal{U}(4.9, 5.1)$	$sub_{1,2}$	✓	
Joint stiffness magnitude at $f = 1$	p	$\mathcal{U}(1.99, 2.01)$N/m	$sub_{1,2}$	✓	✓
Linear joint model gradient	β	$\mathcal{U}(-1.72, -1.68)$	$sub_{1,2}$		✓

Table 7.1: Parameters of the four degree-of-freedom shear structure. The 'true' and simulator columns refer to which numerical models the parameters are used in.

Beam Sub-system

The four stiffness values K from the full-system are affected by the tip stiffness of a cantilever beam K_b. In this case study any of the four beams can be damaged by a midpoint crack x_{cr}, of increasing crack length l_{cr}, as depicted in Fig. 7.3d. In order to illustrate the fact that 'All models are wrong but some are useful' [41] — due to missing physics and/or approximations — the 'true' behaviour and the simulator are derived from different numerical models in the literature. The 'true' behaviour for the change in stiffness from an open crack is formulated using the numerical model defined by Christides and Barr [174] in Eq. (7.5). A bilinear stiffness model, by Sinha et al. [175], forms the simulator presented in Eq. (7.6). Both stiffness models are solved using the Euler-Bernoulli bending beam equation in Eq. (7.7) via numerical integration, where the beam stiffness is calculated via $K_b = -F/y_{tip}$ (y_{tip}[1] is the tip

[1]The notation y here indicates the deflection and not an output.

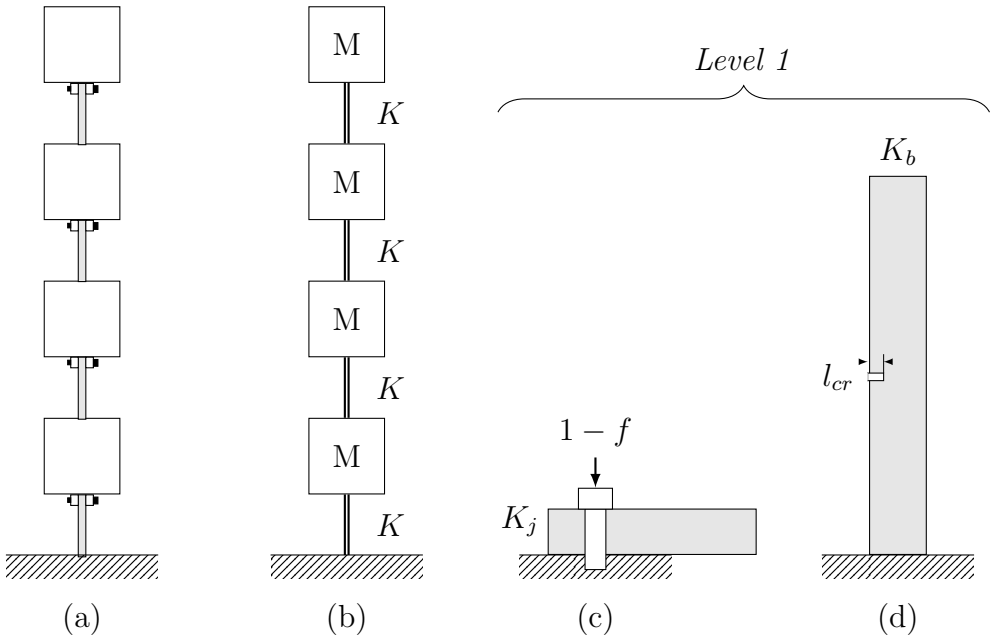


Figure 7.3: Schematics of the four degree-of-freedom shear structure. Panel (a) is the 'true' full-system with beams bolted to the underside of each plate. Panel (b) is the full-system simulator where M is the mass of the blocks and K is the stiffness (a combination of the bolt and beam stiffness in series). Panel (c) is the bolted joint sub-system for a reduction in the non-dimensionalised clamping force f. Panel (d) is the cantilever beam sub-system with a crack of length l_{cr}.

deflection). This means that both the 'true' behaviour and simulator for the beam sub-system map $\boldsymbol{l}_{cr} \rightarrow \boldsymbol{K}_b$. The simulator for this sub-system is the first simulator at level one — denoted $\eta^{sub}_{1,1}$.

$$EI(x) = \frac{EI_0}{1 + C\exp(-2\alpha|x - x_{cr}|/t_b)} \tag{7.5}$$

$$EI(x) = \begin{cases} EI_0 & \text{if } x \leq x_{cr,1} \text{ or } x \geq x_{cr,2} \\ EI_0 - E(I_0 - I_c)\frac{x - x_{cr,1}}{x_{cr} - x_{cr,1}} & \text{if } x_{cr,1} \leq x \leq x_{cr} \\ EI_0 - E(I_0 - I_c)\frac{x_{cr,2} - x}{x_{cr,2} - x_{cr}} & \text{if } x_{cr} \leq x \leq x_{cr,2} \end{cases} \tag{7.6}$$

$$\frac{\partial^2 y}{\partial x^2} = -\frac{M(x)}{EI(x)} \tag{7.7}$$

Where E is the elastic modulus, x is the distance along the length of the beam and M the bending moment. The second moment of areas $I_0 = (w_b t_b^3)/12$ and

$I_0 = w_b(t_b - l_{cr})^3/12$ contribute to the constant $C = (I_0 - I_c)/I_c$, where w_t and t_b are the beam width and thickness respectively. The parameter α is set according to experimental work performed by Christides and Barr [174] (presented in Table 7.1). In the bilinear model, positions beside the crack are calculated by $x_{cr,1} = x_{cr} - l_{eff}$ and $x_{cr,2} = x_{cr} + l_{eff}$ where the effective length of the stiffness reduction due to a crack is $l_{eff} = 1.5t_b$, as defined by Sinha et al. [175]; based on the work by Christides and Barr [174].

The experiment for this sub-system is a static deflection test, due to the quasi-static assumptions in Eqs. (7.5) and (7.6). The beam was forced from $\boldsymbol{F} = \{100, 150, ..., 500\}$N for each crack length $\boldsymbol{l}_c = \{0, 0.1, ..., 0.9\} \times t_b$mm and the tip deflection $\boldsymbol{y}_{tip}$ measured (numerically using Eqs. (7.5) and (7.7)) with observational uncertainty distributed $e^{sub}_{1,1} \sim \mathcal{N}(0, 1^2)$mm. The experimental beam stiffness at the tip was subsequently estimated via the gradient from a least-squares linear regression fit between the force and tip deflection.

The statistical model in the form of Eq. (7.2) can be formulated as in Eq. (7.8).

$$\boldsymbol{z}^{sub}_{1,1}(\boldsymbol{x}^{sub}_{1,1}) = \eta^{sub}_{1,1}(\boldsymbol{x}^{sub}_{1,1}, \boldsymbol{\theta}^{sub}_{1,1}) + \delta^{sub}_{1,1}(\boldsymbol{x}^{sub}_{1,1}) + e^{sub}_{1,1} \tag{7.8}$$

Where $\boldsymbol{z}^{sub}_{1,1}(\boldsymbol{x}^{sub}_{1,1}) = \boldsymbol{K}^{exp}_b(\boldsymbol{l}_{cr})$ (the experimental beam tip stiffness), $\boldsymbol{x}^{sub}_{1,1} = \boldsymbol{l}_{cr}$. The bilinear numerical model forms the simulator $\eta^{sub}_{1,1}(\boldsymbol{x}^{sub}_{1,1}, \boldsymbol{\theta}^{sub}_{1,1})$, where $\boldsymbol{\theta}^{sub}_{1,1} = \{l_b, w_b, t_b, E, x_{cr}\}$ and the output is $\boldsymbol{y}^{sub}_{1,1}(\boldsymbol{x}^{sub}_{1,1}) = \boldsymbol{K}_b(\boldsymbol{l}_{cr})$. Both the observational uncertainty $e^{sub}_{1,1}$ and model discrepancy $\delta^{sub}_{1,1}(\boldsymbol{x}^{sub}_{1,1})$ are assumed unknown. Consequently, a GP regression model is utilised to infer both the model discrepancy and observational noise, regressing from crack lengths $\boldsymbol{l}_{cr}$ to the residual stiffness $\Delta\boldsymbol{K}_b = \boldsymbol{K}^{exp}_b - \boldsymbol{K}^{mode}_b$ where *mode* indicates the simulator output at the modal values of the parameter distributions. Initially a leave-one-out cross validation process was used, where the 50 samples for each crack length were omitted periodically in training. However very small changes in the functional form were witnessed, leading to the full experimental data set being used in training. The bias corrected beam tip stiffness is compared to the simulator output and the experimental results in Fig. 7.4. This demonstrates the ability of a GP regression model to capture the functional form of the discrepancy whilst estimating a homoscedastic observational uncertainty (the results may be improved with a hetroscedastic observational uncertainty model in the GP regression model [138], this is left as an area for further research). The NMSE between the bias corrected and experimental data means are 0.001 showing

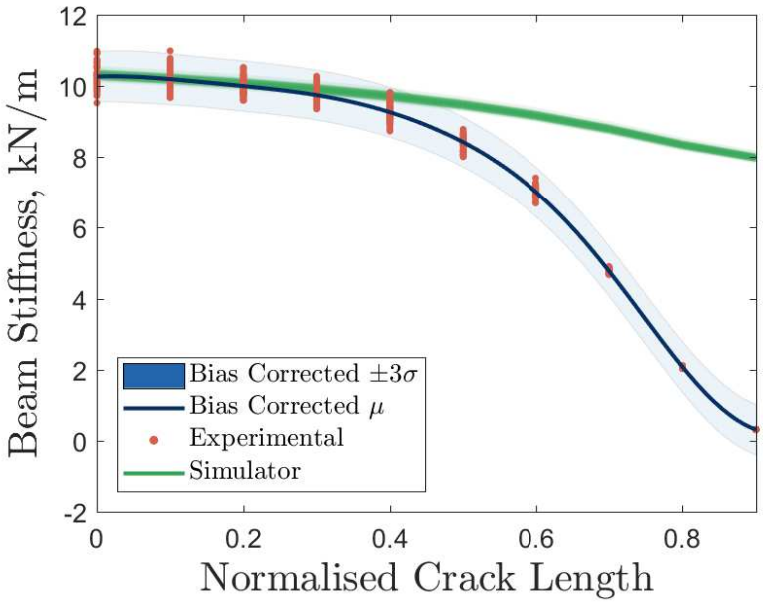


Figure 7.4: Subsystem simulator 1 level 1, $sub_{1,1}$. A comparison of bias corrected, experimental and simulator beam tip stiffness $\boldsymbol{K}_b$ for different crack lengths $\boldsymbol{l}_{cr}$.

excellent agreement.

Bolted Joint Sub-system

The bolted joint stiffness K_j at any of the four locations in the full-system can be damaged via a reduction in the clamping force, parametrised by a non-dimensional clamping force f, presented in Fig. 7.3c. The 'true' behaviour and simulator output of K_j, for a reduction in f, are modelled numerically as shown in Eqs. (7.9) and (7.10) respectively; where Eq. (7.9) is a quasi-static bolt loosening model defined by Todd et al. [176, 177] and Eq. (7.10) is a linear fit of that numerical model. Consequently, the bolted joint sub-system maps $\boldsymbol{f} \rightarrow \boldsymbol{K}_j$. The simulator for this sub-system is the second simulator at level one — denoted $\eta_{1,2}^{sub}$.

$$K_j(f) = K_{ji} \times \tanh\left(\kappa(1-f)\right)\left(p + (1-p)\tanh\left(\kappa\frac{f}{1-f}\right)\right) \tag{7.9}$$

$$K_j(f) = K_{ji} \times (\beta f + p) \tag{7.10}$$

Where K_{ij} is the initial stiffness of the bolted joint, κ adjusts the rate of stiffness change, p adjusts the stiffness function magnitude at $f = 1$ and β is the gradient of the linear model. It is noted that the stiffness function is maximum at $f = 1$ and minimum and $f = 0$ to correspond with a reduction in force, meaning that the outputs for $\boldsymbol{f}$ are reversed.

The experiment for this sub-system is also a static deflection test, due to the quasi-static assumptions in Eqs. (7.9) and (7.10). The joint was forced from $\boldsymbol{F} = \{100, 150, ..., 500\}$N for each reduction in non-dimensionalised clamping force $\boldsymbol{f} = \{1, 0.9, ..., 0.1\}$ and the tip deflection measured (using Hooke's law $F = -K_j y$) with observational uncertainty distributed $e^{sub}_{2,1} \sim \mathcal{N}(0, 1^2)$mm. Again the experimental joint stiffness is estimated from the gradient of a least-squares linear regression model.

The statistical model in the form of Eq. (7.2) can be formulated as in Eq. (7.11).

$$\boldsymbol{z}^{sub}_{2,1}(\boldsymbol{x}^{sub}_{2,1}) = \eta^{sub}_{2,1}(\boldsymbol{x}^{sub}_{2,1}, \boldsymbol{\theta}^{sub}_{2,1}) + \delta^{sub}_{2,1}(\boldsymbol{x}^{sub}_{2,1}) + e^{sub}_{2,1} \tag{7.11}$$

Where $\boldsymbol{z}^{sub}_{2,1}(\boldsymbol{x}^{sub}_{2,1}) = \boldsymbol{K}^{exp}_j(\boldsymbol{f})$ (the experimental joint stiffness) and $\boldsymbol{x}^{sub}_{2,1} = \boldsymbol{f}$. The simulator $\eta^{sub}_{2,1}(\boldsymbol{x}^{sub}_{2,1}, \boldsymbol{\theta}^{sub}_{2,1})$, is a linear numerical model (Eq. (7.9)) where $\boldsymbol{\theta}^{sub}_{2,1} = \{p, \beta, K_{ji}\}$ and the output is $\boldsymbol{y}^{sub}_{2,1}(\boldsymbol{x}^{sub}_{2,1}) = \boldsymbol{K}_j(\boldsymbol{f})$. Using a GP regression model, both the observational uncertainty $e^{sub}_{2,1}$ and model discrepancy $\delta^{sub}_{2,1}(\boldsymbol{x}^{sub}_{2,1})$ are inferred in the same manner to the beam sub-system, regressing between $\boldsymbol{f}$ and $\Delta\boldsymbol{K}_j = \boldsymbol{K}^{exp}_j - \boldsymbol{K}^{mode}_j$. Similarly to the beam sub-system an initial leave-one-out cross validation process was employed, and due to relatively small changes in the functional form the full experimental data set was used in training. Fig. 7.5 shows a comparison for the bias corrected simulator and experimental joint stiffness for a reduction in clamping force. As with the beam sub-system, the bias corrected joint stiffness captures the functional form of the model discrepancy, which is not captured by the simulator. Homoscedastic assumptions in the GP regression model again mean that a homoscedastic observational uncertainty is inferred. The NMSE between the bias corrected and experimental data means are 0.048 showing good agreement.

Full-System Integration

The full-system is an undamped linear spring-mass system (Fig. 7.3b) where the spring stiffnesses K are composed of the beam tip stiffness and bolt stiffness in series

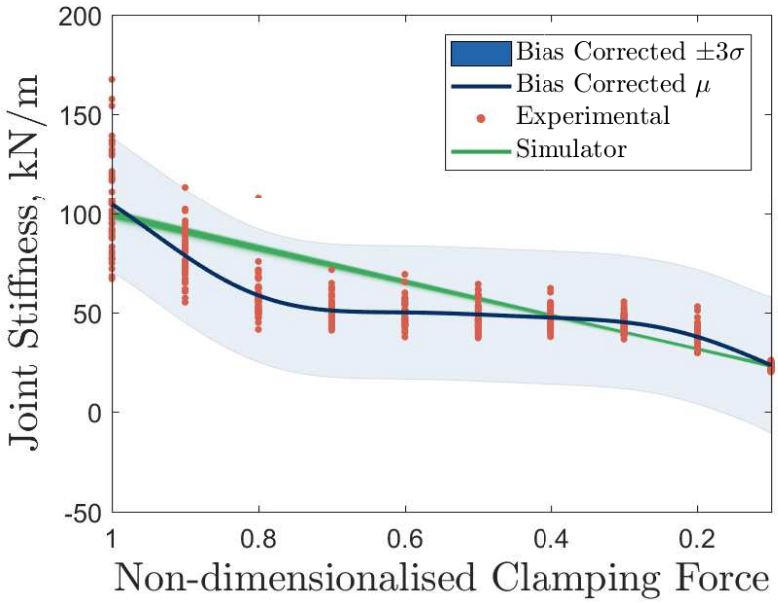


Figure 7.5: Subsystem simulator 2 level 1, $sub_{2,1}$. A comparison of bias corrected, experimental and simulator joint stiffness $\boldsymbol{K}_j$ for a reduction in non-dimensionalised clamping force $\boldsymbol{f}$ - it is noted that the axis is reversed as [176, 177].

$K = (K_b \times K_j)/(K_b + K_j)$. The four degree-of-freedom system can be solved via an eigenvalue problem for the natural frequencies of the system. For this case study the 'true' and simulator numerical models are equivalent with the only difference being the input values for the beam tip and joint stiffnesses under damage. This means that there is no model discrepancy at the full-system level resulting in Eq. (7.12); due to the assumption that all the model discrepancy due to damage can be captured at a sub-system level. The percentage difference of natural frequency $\Delta\boldsymbol{\omega}_n$ is used as a damage feature in this case study, as it is a more damage sensitive feature compared to natural frequency [178].

$$\boldsymbol{z}^{full}(X^{full}) = \eta^{full}(X^{full}, \boldsymbol{\theta}^{full}) + e^{full} \tag{7.12}$$

Where $\boldsymbol{z}^{full}(X^{full}) = \Delta\boldsymbol{\omega}_n(X^{full})$ — the percentage differences of the experimental four natural frequencies under damage. There are several inputs to the full-system $X^{full} = \{\boldsymbol{n}_{cr}, \boldsymbol{n}_f, y^{sub}_{1,1}(\boldsymbol{x}^{sub}_{1,1}), y^{sub}_{2,1}(\boldsymbol{x}^{sub}_{2,1})\}$ where $\boldsymbol{n}_{cr}$ and $\boldsymbol{n}_f$ indicate the floor in which the damage occurs, for the crack and loosened bolt respectively. As $\boldsymbol{n}_{cr}$ and $\boldsymbol{n}_f$ are only position inputs for this simple system it is assumed that no input dependent model discrepancy term is required. The other inputs are: $y^{sub}_{1,1}(\boldsymbol{x}^{sub}_{1,1}) = \boldsymbol{K}_b(\boldsymbol{l}_{cr})$,

— the beam tip stiffness under a midpoint crack — $y_{2,1}^{sub}(\boldsymbol{x}_{2,1}^{sub}) = \boldsymbol{K}_j(\boldsymbol{f})$, — the joint stiffness from a reduction in clamping force (dependent on their sub-system model discrepancy). The full-system simulator $\eta^{full}(X^{full}, \boldsymbol{\theta}^{full})$ also depends on the parameter set $\boldsymbol{\theta}^{full} = \{l_b, w_b, t_b, l_p, w_p, t_p, E, \rho, K_{ij}\}$. The numerical nature of this case study allows the comparison of both 'true' and simulator outputs, $\Delta\boldsymbol{\omega}_n$, under these damage types.

500 Monte Carlo realisations, drawn from the outputs of level one and the full system parameters were generated in order to compare the bias corrected, simulator and 'true' full-system outputs. Figures 7.6 and 7.7 present the bias corrected $\Delta\boldsymbol{\omega}_n$ under increasing crack length and for a reduction in clamping force. As expected, the increase in crack length has a greater effect on the natural frequencies of the system compared to a reduction in clamping force (reflected in the stiffness reductions in Figs. 7.4 and 7.5). The first natural frequency is the most affected by an increase in crack length, with a comparable reduction in the first, second and third natural frequencies for a reduction in clamping force. Figure 7.8 demonstrates an example comparison of the output distributions for $\Delta\omega_1$, where both damage types are located at the first floor and the only damage type is an increase in crack length. A visual comparison shows that for the first five damage states $\boldsymbol{l}_{cr} = \{0, 0.1, 0.2, 0.3, 0.4\} \times t_b$, the distributions are very similar, after which the simulator fails to capture the correct distribution forms, whereas the bias corrected simulator maintains a good fit.

Hypothesis testing using the KS-two sample test were performed at each damage scenario, for each natural frequency, totalling 6400 combinations. The significance level, the upper bound of the probability of type 1 errors, was $\alpha_H = 0.01$. The percentage of null hypotheses H_0, that were not rejected, for both the bias corrected and simulator outputs at a full-system level, when compared to the 'true' outputs, were 97.5% and 26.2%. This demonstrates that the proposed subfunction discrepancy approach outperforms utilising the simulator without recognition of model discrepancy, providing a significant improvement. In order to illustrate this further Fig. 7.9 presents a comparison between the subfunction discrepancy technique and the original simulator for the first natural frequency; this was the worst performance of the uncertainty integration strategy. It can be seen that the subfunction discrepancy methods performance is the same for all locations at 90% (in other natural frequencies there are differences at different locations). The original simulator however, performs best when damage is located at the highest floors, at 54%. This is because as damage is located at a lower floor it will have a greater effect on the first natural frequency

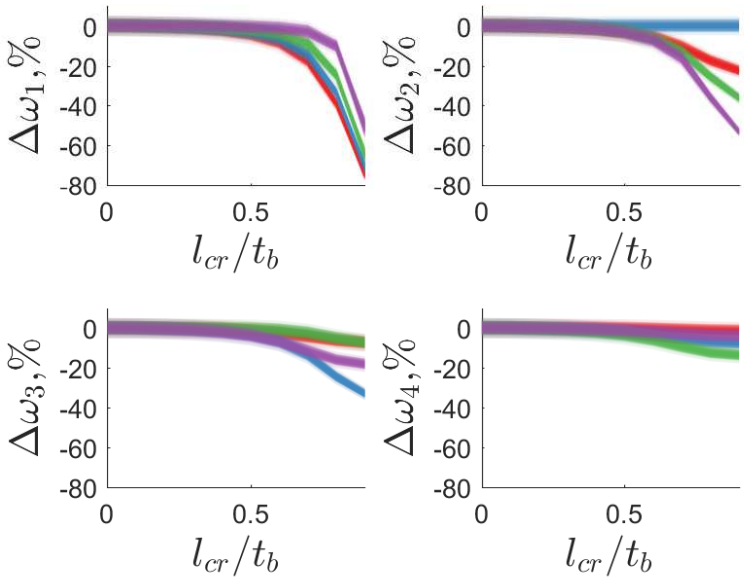


Figure 7.6: Bias corrected outputs — the percentage difference of the four natural frequencies $\Delta\boldsymbol{\omega}_n$ — for an increase in crack length (at the midpoint) at different floors of the full-system $\boldsymbol{n}_{cr} = \boldsymbol{n}_f = \{1, 2, 3, 4\}$ (red, blue, green, purple).

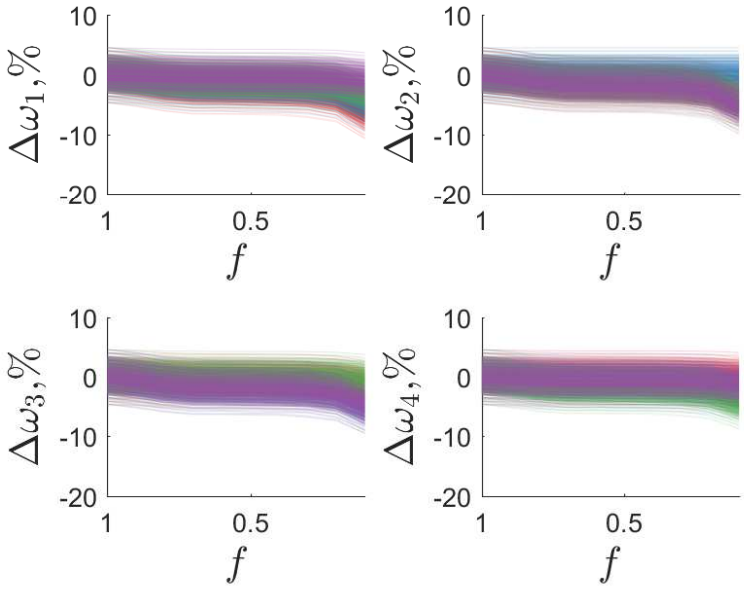


Figure 7.7: Bias corrected outputs — the percentage difference of the four natural frequencies $\Delta\boldsymbol{\omega}_n$ — for a reduction in clamping force at different floors of the full-system $\boldsymbol{n}_{cr} = \boldsymbol{n}_f = \{1, 2, 3, 4\}$ (red, blue, green, purple). It is noted that the x-axis is reversed as [176, 177].

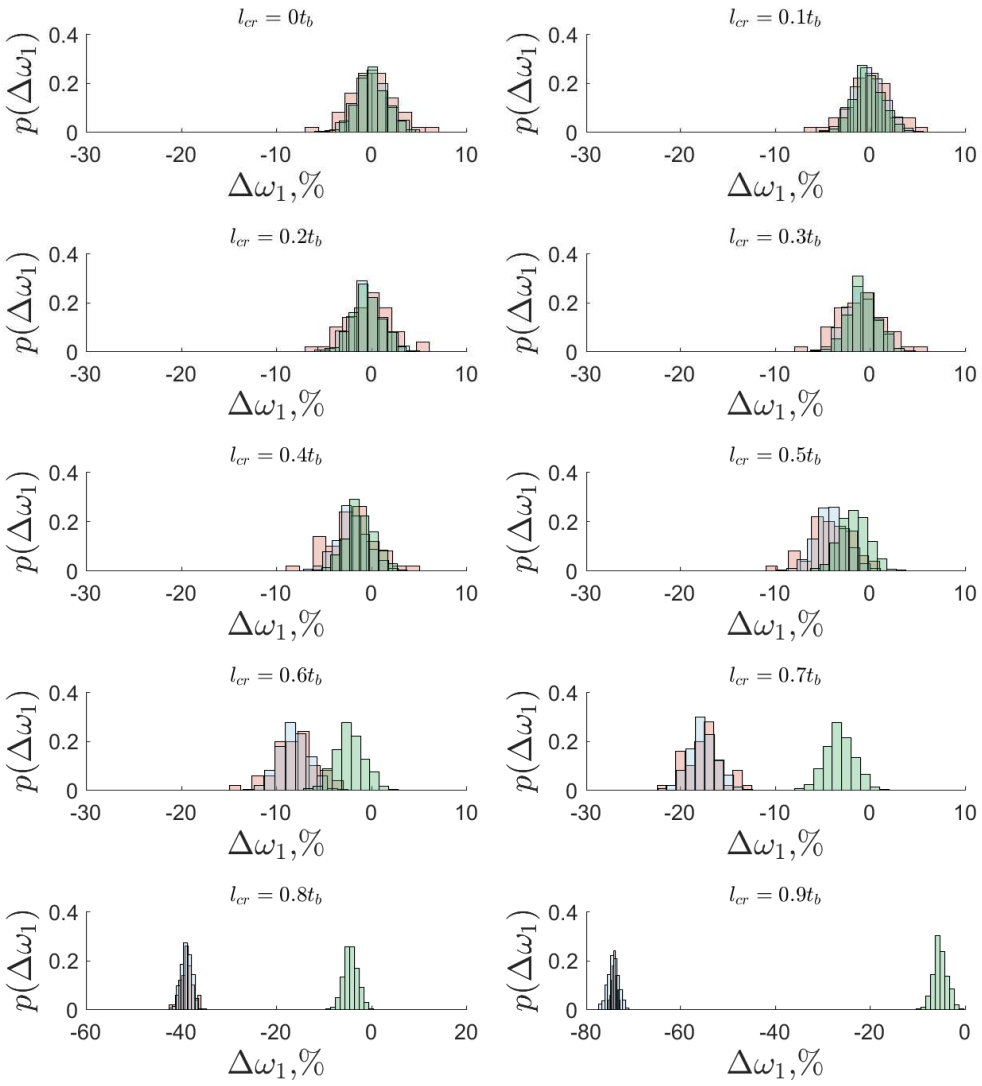


Figure 7.8: A comparison of $\Delta\boldsymbol{\omega}_1$ for increasing crack lengths at the midpoint of floor one; original simulator (green), subfunction discrepancy technique (blue) and 'true' (red) outputs. It is noted that the axis limits are different for $\boldsymbol{l}_{cr} = \{0.8, 0.9\} \times t_b$ due to the large decrease in natural frequency.

of the system (as it is the first bending mode). Damage due to a crack located at the lower floors also affects the first natural frequency more than damage at the joint. This is expected from Figs. 7.4 and 7.5; both indicate the original simulator fails to capture the stiffness reduction for a crack to a greater extent than the due to a reduction in clamping force. The NMSEs for all 6400 combinations for both

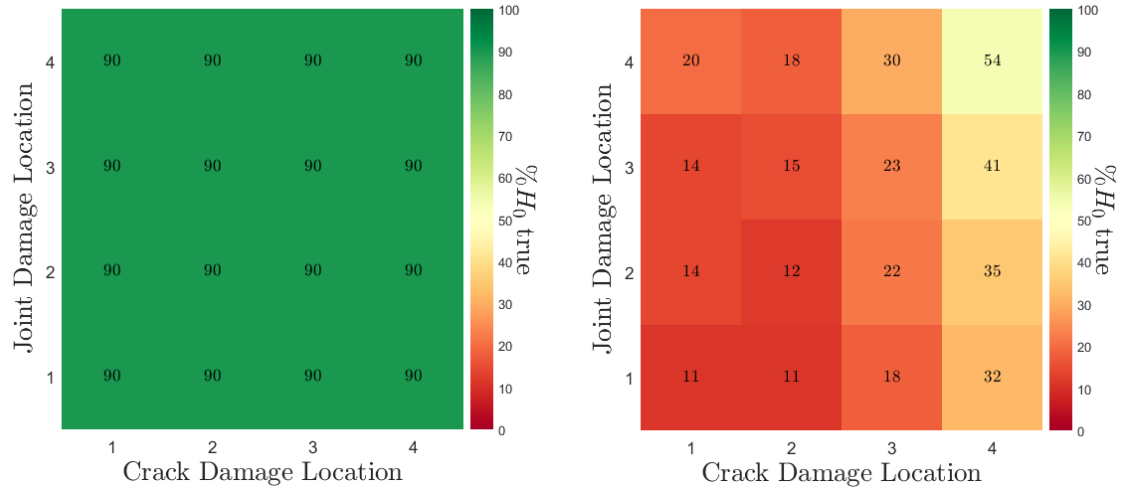


Figure 7.9: The percentage of null hypotheses H_0, that are not rejected for the first natural frequency. Panel (a) presents the results for the subfunction discrepancy approach and panel (b) for the original simulator.

the original simulator and subfunction discrepancy approach were 93.57 and 0.002 respectively. This highlights the inability to capture the mean trend in the original simulator, and the excellent agreement in the mean outputs for the subfunction discrepancy technique and 'true' full-system.

In order to analyse these results further, Fig. 7.10 presents a comparison of hypothesis test outcomes from all combinations of damage located at floor one, for the first natural frequency, are presented. This is chosen as the original simulator performs worse at this location, aiding the diagnoses of the difference in performance. The null hypothesis is rejected for all clamping force reductions when the crack length is at 90% of the beam thickness for the subfunction discrepancy method. This indicates that the beam tip stiffness for this crack length has not accurately been captured, and should be an area of model improvement at the sub-system level. On the other hand, the original simulator fails to capture the majority of damage scenarios. It performs best when the crack length is small (under 20% of the beam thickness) and when the reduction in clamping force results in the linear model overlapping the hyperbolic tangent model (Fig. 7.5). Consequently, the failure to adequately capture the model form at a sub-system level will result in poor full-system performance and as all physics can never be fully captured in any model, a mechanism for quantifying the functional form and uncertainty due to model discrepancy is paramount.

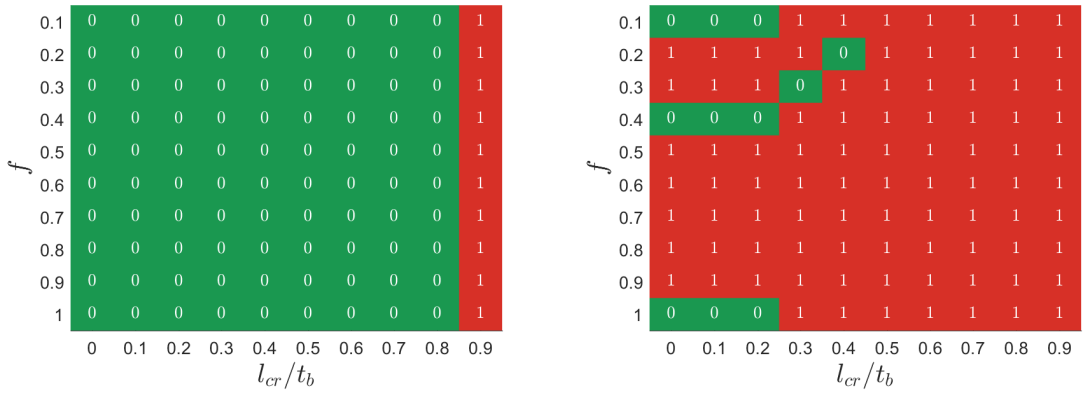


Figure 7.10: KS two sample hypothesis test results for the first natural frequency where both damage types are located at floor one; panel (a) is the subfunction discrepancy approach and panel (b) the original simulator.

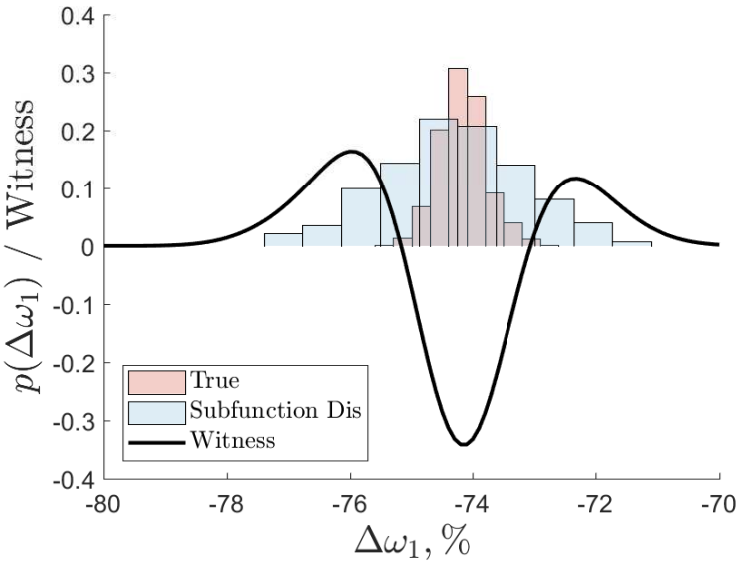


Figure 7.11: A comparison of the histograms and witness function of $\Delta\boldsymbol{\omega}_1$ for $l_{cr} = 0.9 \times t_b$ at floor one.

To illustrate reasons for the results in Fig. 7.10, Fig. 7.8 presents a comparison of the output distributions from the 'true', original simulator and subfunction discrepancy method. The distributions are of $\Delta\boldsymbol{\omega}_1$ when both damage types are located at floor one, where $f = 1$, and crack length is increased. Figure 7.8 shows that for $\boldsymbol{l}_{cr} = 0, 0.1, 0.2 \times t_b$ all three distributions are overlaid with little difference in the probability mass. As the crack length increases further, a shift in the output mean of the original simulator occurs in Fig. 7.8, leading to a rejection in the null hypothesis.

The subfunction discrepancy technique provides a good fit with the 'true' full-system until $l_{cr} = 0.9 \times t_b$, where the true system has a smaller variance — the percentage difference of the mean and variance at this damage extent are 0.2% and 737.4%. These differences are highlighted in Fig. 7.11, where the witness function, the difference between two kernel mappings of the samples [45, 131], is presented. The witness function was generated using a Gaussian kernel where the scale parameter σ, was set using a median heuristic. The witness function provides a visual illustration the offset in mean and smaller variance.

7.4 Conclusion

A multi-level uncertainty integration strategy based on a subfunction discrepancy approach has been demonstrated. The strategy seeks to divide a structure into different levels and sub-systems where validation of damage mechanisms is achievable. At each of these sub-systems model discrepancies are inferred in order to improve the predictive capability of the simulator. Finally, the sub-system model discrepancies and parameter uncertainties are propagated through to the full-system level providing confidence in predictions at a full-system level.

A numerical case study has been presented demonstrating the technique on a simple four degree-of-freedom shear structure. The objective of this study was to predict damage sensitive features — percentage differences of natural frequencies — when two types of damage were introduced to the structure, namely a midpoint crack of increasing length and a reduction in clamping force at the joint for various positions in the structure. The full-system was divided into one level where there were two sub-systems: a beam sub-system and a joint sub-system. At each of these sub-systems experimental data was generated from the 'true' process with the addition of observational uncertainty and model discrepancies inferred between simulator outputs and the experimental data. The uncertainties and model discrepancies were propagated to the full-system level where, due to the numerical nature of the case study, a comparison was made with the 'true' outputs. Hypothesis testing was performed on the output distributions for the 6400 combinations of inputs. This demonstrated that the multi-level uncertainty strategy had improved the predictive performance from 26.2% to 97.5% (of not rejecting the null hypotheses). In terms of mean predictions, the strategy improved NMSEs from 93.57 to 0.002. The enhanced

predictive capability clearly indicates the benefits of this approach for the case study.

Further research should be conducted into applying the methodology to a real world case study and for different types of simulator and model fidelities. The approach should also be applied with different divisions of a structure in order to quantify the differences in outputs due to the initial set up of the technique. Additionally, demonstration of the process with calibration included is left as further work. Finally, the inclusion of hetroscedastic GPs to better capture the model discrepancies from hetroscedastic experimental data should be pursued.

Chapter 8

Discussion and Conclusions

This thesis was motivated by the development of an alternative model-based approach to SHM whereby simulators were utilised in a forward manner in generating training data for machine learning models, defined as forward model-driven methods. The aim of establishing this framework was to provide solutions to current issues with both data-driven and model-driven strategies, namely the lack of available damage state data and inferring the health state from updated parameters, as presented in Chapter 1.

Chapter 2 proposed a framework for forward model-driven SHM constructed from two key components: developing a simulator that is capable of predicting outputs for various health states that are statistically representative of those obtained from in-service data, and defining a machine learning methodology that robustly infers decision bounds between health states. This emphasis on generating simulators that produce representative predictions highlighted three key challenges to realising forward model-driven SHM, and hence were the focus of this research.

Firstly, a clear definition of what a valid predictive output is must be established, that is, a clear quantification that the simulator is adequate for training a robust decision bound within the chosen machine learning method. Chapter 3 sought to tackle this problem, outlining that a valid prediction would be one in which a simulator output could be considered to be from the same statistical distribution as the in-service data. As a consequence a validation strategy that considered probabilistic prediction was developed, involving hypothesis testing, statistical distance metrics, such as

distribution based distances, visual diagnostics, e.g. witness functions, and where appropriate any deterministic metrics. The components of this strategy aimed to diagnose issues within the predictive distributions and could be used to aid decisions on whether model selection, new test strategies or simulator evaluations were required.

Secondly, the simulator predictions must account for uncertainties introduced by model discrepancy. This was deemed important in realising valid predictions from simulators as without corrections for model form errors caused by missing physics, simulator outputs could be far those observed. In addition, calibration without considering model discrepancy will lead to bias in the inferred parameter set, and will have no guarantees in identifying the 'true' parameter distribution. This becomes especially problematic if these identified parameters are incorporated in a multi-level uncertainty integration strategy. Two techniques were investigated that incorporate mechanisms which account for model discrepancy, by inferring its functional form, namely BCBC and BHM in Chapters 5 and 6 respectively. The two approaches provide different frameworks for calibration, with BCBC attempting a Bayesian solution whereby the posterior parameter distribution and model discrepancy are jointly inferred. In contrast BHM, an approximate Bayesian method (a sub-class of ABC), separates out the inferences, with the approximate posterior parameter distribution inferred by assuming a uniform additive model discrepancy. Subsequently, model discrepancy can be inferred based on a proposed importance sampling methodology which is akin to performing Bayesian model averaging. When the assumption that model discrepancy is uniform and additive holds, the BHM-importance sampling approach offers a solution to the non-identifiability problems associated with BCBC by decoupling the inference process.

Thirdly, validation without obtaining full-system level data must be achieved. This is a significant challenge in making forward model-driven methods a solution to the lack of available full-system damage data problem. In order to approach this issue, an investigation into multi-level uncertainty integration via a subfunction discrepancy approach was undertaken in Chapter 7. The developed technique divides a structure into levels of sub-systems that capture the relevant damage mechanisms of interest. The assumption is that by capturing the damage physics at these sub-system levels, calibration and model discrepancy inferences can occur based on more easily obtained experimental data. Once calibrated the model form corrections and uncertainties can be propagated through to the full-system, where the validated damage functions are assumed to hold. By separating out the inferences to each sub-system the problem

becomes more computationally manageable, and allows for individual sub-systems to be recalibrated, without performing inference at all levels and sub-systems if new data is obtained.

These technologies and strategies provide potential solutions to the aforementioned main challenges for realising forward model-driven SHM. The sections below present details of the important conclusions from the work presented in this thesis. Furthermore, limitations of the methodologies defined within each chapter are discussed before future work is outlined. Finally, the author's opinion about the future directions for forward model-driven SHM are described.

8.1 Conclusions

As highlighted, validation is perhaps the most challenging aspect of gaining confidence in an SHM system. It becomes increasingly important when simulator predictions are the only evidence for patterns between health states. Without valid predictions the inferred classifier would be extremely dangerous to implement on any real world structure, as there would be no confidence that the labels are assigned based on the real system's damage physics. Chapter 3 sought to define within the context of forward model-driven SHM a validated simulator. This definition revolved around obtaining an understanding of adequacy for the uses of the simulator predictions, specifically that of defining decision bounds. When incorporated in the training of machine learning methods any differences in the key statistical moments, such as offsets in the mean, inflation or under-estimate of the variance, etc. could lead to confusion or inappropriately specified decision bounds. Consequently, a validated simulator was established to be one in which its prediction could be determined to be from the same statistical distribution as in-service data. This led to the development of a validation strategy in which hypothesis testing, quantification using metrics that consider full distributions, visualisation tools and deterministic metrics were incorporated. These tools aimed to diagnose issues within the predictive distributions and could be used to aid decisions on whether model selection, new test strategies or simulator evaluations were required.

Hypothesis testing methods were presented using both KS-, MMD- and Bayesian hypothesis tests. In addition, distribution-based distance metrics were defined, categorising existing validation metrics — such as the area metric — within the statistical

distances terminology. Through numerical examples it was demonstrated that the total variation distance was most sensitive to differences between distributions, followed closely by the Hellinger and MMD distances. KL-divergence was found to be difficult to interpret but relatively effective in determining whether replication of a target distribution with a proposed alternative was impossible. The area metric was found not to be particularly sensitive to differences between distributions. However, the area metric has the same units as the quantity being analysed, and for this reason is particularly useful. Furthermore, the Kolmogorov distance is mostly sensitive to changes in the central probability mass, meaning that KS-tests are suboptimal when differences in the distributions are contained within the tails. Moreover, MMD two samples tests provide a non-parametric method for comparing two sets of samples and are sensitive to multiple types of differences between distributions, making this test more robust in most applications. More than that, MMD provides a method for visually determining the differences between two distributions through the witness function. This offers a powerful method for interrogating a simulator predictions validity.

Many of the techniques incorporated within a forward model-driven strategy involve interrogating a simulator over a large parameter domain. Given that simulators are often computationally expensive to evaluate, cheap surrogate emulators were investigated in Chapter 4. These technologies are vital in making the statistical methods presented in this thesis computationally practical. The merits and disadvantages of several emulator constructions were discussed, culminating in the selection of GP emulators as a robust tool for developing surrogate models. This was due to GPs containing built in regularisation, within the formulation of the marginal likelihood, preventing the tool from overfitting. In addition, the Bayesian formulation provides a quantification of code uncertainty, an important characteristic in determining the emulator performance and information about where simulator evaluations may aid inference of the underlying function.

When emulating a deterministic simulator, emulator predictions should replicate known simulator outputs with no code uncertainty. Mathematically a GP will achieve this requirement. However, practical implementation can lead to poorly conditioned covariances matrices within the inference process, causing either the covariance not to be invertible or numerical instabilities to occur in the inversion. The addition of a nugget term was therefore implemented to resolve these numerical issues, although it is recognised that this may lead to type-II MLE solutions being

the most likely. In light of this, a penalty term was introduced, removing the type-II mode of the NLML and forcing the emulator to fit known simulator evaluations exactly. Diagnostics and validation tools were defined for GP emulators, ensuring that the model form and inferred parameters were appropriate. A numerical example was demonstrated in Chapter 4 in order to demonstrate the use of these diagnostics. Another implementation consideration is how to generate training data from simulator evaluations that will aid the inference of the GP emulator. GMLHCs were introduced offering a methodology for generating computer DoEs that reduce the emulator's code uncertainties near the edge of the parameter domain. This approach was shown to be more effective than alternative LHC-based designs.

The formulation of GP emulators in scenarios where the parameter space was large, or time series data was part of the training set, were investigated. In these scenarios the number of training points may become large even for a small number of simulator evaluations. A comparison of sparse GP formulations and their applicability to surrogate modelling was investigated. Two categories of approach were compared — model and posterior approximations — where it was demonstrated on a numerical example that model approximations lead to overfitting issues. PEP formulations were found to be a universal framework for considering model and posterior approximations, with the limits being the FITC and VFE approaches. Concerns were raised about the practicalities of sparse methods in performing emulation. These concerns arose from the fact that a noise term is incorporated as part of the posterior approximation formulations, leading to type-II solutions. This results in GP predictions not reproducing known simulator evaluations exactly and with no code uncertainty. Finally, other GP extensions for multiple output, stochastic simulators and dynamic processes were described, showing the potential improvements to the GP emulator framework.

Chapters 5 and 6 describe two methods — BCBC and BHM — for calibrating simulators when model discrepancy is present and inferring its functional form. BCBC utilises two GP models in order to jointly perform Bayesian inference on both the estimate parameters and the model discrepancy term. An issue with the approach is that the prior assumption that the model discrepancy is distributed as a GP proves to be too flexible when part of a joint inference process with the parameters. This causes non-identifiability issues, leading to a lack of confidence in the inferred parameter distribution. In forward model-driven SHM the method may be applicable when only forward predictions are required, and the inferred parameters

are not utilised. In this scenario the corrected output predictions would be useful in the validated domain, but the simulator could not be trusted to extrapolate. BCBC was applied to two representative building structures in which it was found to have adequate performance, outperforming Bayesian calibration (without bias correction) for a three storey representative building structure. Furthermore it was found that both Gauss-Hermite quadrature and adaptive Metropolis MCMC provided similar predictive results and inferred posterior distributions, where Gauss-Hermite quadrature is computationally more efficient in low dimensional problems.

BHM provides an approximate Bayesian method for calibrating simulators whilst accounting for model discrepancy. The rejection based technique removes parts of the input space based on implausibility metrics. By construction these metrics incorporate the possibility of model discrepancy via an additional variance term. The method will perform exact Monte Carlo inference given an additive uniform model discrepancy, given that it is a sub-category of ABC. Approximate posterior parameter distributions have been demonstrated to be obtainable through importance sampling. This means that inference of the parameter distribution can be performed separately to that of the model discrepancy, unlike BCBC, improving non-identifiability issues.

As BHM is performed in an iterative manner, where the simulator is evaluated at new locations until the code uncertainty of the emulator is below the prior observational and model discrepancy uncertainties, sequential design methods can be incorporated into the framework. Here two heuristics were developed, namely probability of non-implausibility and expected (un)improvement. It was found that probability of non-implausibility exploited known non-implausible locations well, but failed to efficiently explore the full parameter domain. Expected (un)improvement on the other hand had a better balance between exploitation and exploration. Alternative sequential methodologies using information-based metrics were discussed but left as an area of further research. In addition, within the BHM framework the parameter domain is sampled. Although not explored in this thesis, improved sampling methods, such as an SMC-BHM methodology could allow more efficient sampling of this parameter domain.

A further extension to the BHM methodology was proposed, inferring functional model discrepancy uncertainties using importance sampling. Although specifically applied to the BHM parameter posterior distributions this could in theory be applied to any scenario where the simulator parameter distributions are known. The technique, akin to Bayesian model averaging of multiple GP regression models, can be

applied with both MLE estimates and marginalisation of the GP hyperparameters. It was demonstrated that although the empirical Bayes form provided closer predictive distributions to that of the dual-importance sampling technique, the reduced uncertainty in the hyperparameters may lead to overconfidence in certain scenarios. These methods were applied to a representative building structure whereby the mean predictions were shown to be in agreement with the observational data. Unfortunately the rigorous handling of the model discrepancy and hyperparameter uncertainties led to increased predictive variance. Despite this problem, the technique can be used to inform improved model selection and with iterations of this process may lead to valid predictions. Furthermore, the case study used three samples in the inference stage and with more data the predictive distributions should improve.

A considerable difficulty in developing a forward model-driven SHM strategy is establishing a technique for producing validated predictions when full-system health state data is unavailable. Chapter 7 aimed to investigate this issue through the development of a multi-level uncertainty integration technology. As a result a subfunction discrepancy technique was developed and applied to a numerical shear structure. The approach seeks to divide the full-system mathematically into subfunctions where calibration can be performed, identifying the uncertainties and model form errors at these sub-system levels. The method assumes that the physics governing the introduction of damage can be adequately captured at a sub-system level, and when propagated through to a full-system level produce valid predictions. The subsequent application of the method to a shear structure under two sources of damage, modelled by two sub-system simulators, demonstrated the potential for this technique. 97% of the predictive distributions for the 6400 combinations were valid when the subfunction approach was applied, this is in contrast to just 26.2% when model discrepancies were not considered.

8.2 Limitations

Several limitation have been established through the course of this thesis. Firstly, it was observed that validation, even when expressed through quantified objective metrics, is subject to a degree of subjectivity. Ultimately it is up the modeller, experimentalist and those with interests in the SHM system to decide upon their interpretation of results. Ideally an independent body would determine, based on

validation metrics, the validity of the predictions.

Even when the objective is to create predictions from the same underlying distribution as observational data several limitations arise. Firstly, one must obtain enough observational data as to have adequate information to define the underlying distribution. Rarely in SHM will this be the case, even at a sub-system level. This means that validation statements are only as good as the statistical interpretation of the data, and if the observation set contains multiple outliers, incorrect conclusions could be drawn. Moreover hypothesis testing, in a frequentist sense, requires a definition of statistical significance. Within the hypothesis testing literature there is much debate on how to set this value, as well as the general usefulness of hypothesis testing for determining whether two samples are drawn from the same distribution. Linked to these problems are the difficulties in evaluating statistical distances. Most of the formulations outlined in this thesis require knowledge of density functions, which may not always be known. Non-parametric distances such as the MMD distance provide a degree of solution, however, in practice the RKHS embedding requires numerous samples in order to encode an accurate representation. Furthermore, interpretation of distances, even when bounded to a unit interval can be challenging, causing additional subjectivity in establishing whether an output is valid.

GP emulators are identified based on a set of training data. This set must be representative of the underlying function that is to be modelled. Problems may arise in certain scenarios, for example where the function is sub-sampled, and these will often produce a model that fails to capture the underlying state. A thought experiment can be conducted to verify this outcome. Imagine the simulator is a sinusoidal function at a single frequency, where evaluations have been obtained at that frequency. This would lead to the simulator producing the same output for each run. Given these evaluations the only reasonable model to construct is a horizontal constant prediction. This would therefore fail to capture the 'true' underlying function, which is a sinusoid. This is a general problem for any black-box emulator, but also applies to GPs. Solutions to this problem are to collect a more representative training set, or to incorporate any knowledge about the expected functional form into the mean and covariance functions. However, these solutions may not always be practical, i.e. it is hard to determine if the training set is truly representative, and limited knowledge of the simulators functional form are often know prior to obtaining multiple evaluations. Furthermore, diagnostic metrics will only identify these problems if the information is contained within the validation

set, which again may not be the case. Limitations also arise in implementing sparse GP formulations as emulators. The key issue is that posterior approximations, although not susceptible to overfitting, incorporate a noise component as part of the approximation. This will lead to known simulator evaluations not being fitted exactly in cases with no code uncertainty.

Non-identifiability problems are a challenge to all calibration methods. BCBC suffers particularly from these issues when the prior for the model discrepancy is an unconstrained GP. The GP assumption is generally too flexible for most applications and will mean that numerous parameter solutions can become more likely than is realistic. BHM, although removes this particular problem, is founded on the assumption that model discrepancy is uniform. This is often not appropriate for most applications, meaning that there are no guarantees that the inferred approximate parameter posteriors are not biased. In addition, model discrepancy inference via importance sampling involves constructing multiple GP models. This may become too computationally expensive in applications where their are a high level of observations — although this is not expected to be the case in most SHM scenarios.

Central to the subfunction discrepancy approach is the assumption that a structure can be divided into sub-systems where the damage physics can be captured in isolation to the remaining structure. This may be too strong an assumption in most scenarios. The process also relies on being able to obtain data at each of these sub-systems, and that these divisions are practically achievable. Furthermore, interactions between sub-systems may be too challenging to model at all, and may lead to large model discrepancy uncertainties. When this is the case extrapolation will be problematic as the inferred GP models will return to the prior outside of the training data.

8.3 Future Work

Several areas of further work are envisaged both as solutions to the limitations and as general extensions to the described approaches. Firstly, the aforementioned technologies should be incorporated in a complete implementation of forward model-driven SHM and rigorously compared to both state-of-the-art inverse model-driven and data-driven techniques. It is hoped that forward model-driven SHM will provide similar results to a data-driven method and outperform inverse model-driven approaches.

Within the validation work, further research should be conducted into other hypothesis testing and statistical distance metrics, and their applicability to validating probabilistic simulator predictions. In particular it has been highlighted that the KSD, a non-parametric one-sample test, should be investigated.

Various GP technologies were outlined within Chapter 4 that were not investigated. In particular multivariate GP formulations would offer better prior assumptions for most multiple output simulators; as often the outputs are mathematically correlated. Dynamic GP formulations may also prove useful when emulating and calibrating dynamical system. Furthermore, methods for determining how representative training data is of the underlying simulator should be investigated, helping to resolve the limitation previously mentioned.

Constraints, and better prior representations when modelling the model discrepancy as a GP should be considered. This would help resolve the non-identifiability problems within BCBC but would also improve model discrepancy inference in BHM and the subfunction discrepancy approach. All these methods should be applied to more complex case studies in order to evaluate their effectiveness when a high dimensional parameter space and complex model discrepancy exist.

Several extensions to BHM should be investigated. These include improving the parameter domain sampling technique by incorporating knowledge from previous waves, potentially in an SMC-BHM formulation. Additionally, entropy-based sequential designs should be employed within the BHM framework. This would provide a more rigorous methodology for determining simulator evaluations.

There are more potential approaches to the multi-level uncertainty integration problem. The subfunction method should be applied to a real world case study, and a more general methodology for sub-system division identified. Finally, hetroscedastic GP formulations should be implemented, where it is expected that model discrepancy inferences and therefore full-system predictions will improve with more accurate estimations of the feature uncertainty.

8.4 Future Directions for Forward Model-Driven Structural Health Monitoring

Forward model-driven SHM has been demonstrated to have significant potential in solving the drawbacks in both inverse model-driven and data-driven SHM. Although this thesis outlines several technologies for realising forward model-driven SHM, future work is still required. Key topics to be investigated are developing more accurate damage models and establishing a robust model selection framework in which information from model discrepancy inference can be used to improve model form errors. The generation of simulators also provides a mechanisms for identifying damage sensitive features. This is a large area of further research that will also aid all approaches to SHM. More work is required in developing monitoring system design based on these validated simulators.

Potential new approaches to semi-supervised learning are also available, with simulators providing a method of generating labels without need for direct supervision. Likewise, Bayes risk approaches that are formulated specifically based on feature vector distributions from simulators should be developed; providing an informative method for communicating probabilistic health decisions to asset managers.

Not addressed in this thesis are how decisions in the framework should be made, particularly when to obtain more validation data, when to perform more simulator evaluations and when model selection should be pursed. Defining these processes based on the reduction of uncertainty and increase in predictive performance should be investigated.

Finally, future work should continue to purse rigorous and robust methods for generating full-system simulator outputs that are valid, based only on data sources that do not include those at a full-system level. Multi-level uncertainty integration is a challenging task and will require further developments in order for approaches to be generally applicable across any structure without complex and bespoke construction.

The proposed forward model-driven framework is a promising approach to SHM, which may not only provide solutions to issues associated with current methodologies, but may also help improve those existing technologies in identifying damage sensitive feature and creating monitoring systems that maximise the probability of detecting damage. It is hoped that this category of approach to SHM receives wider

investigation in the SHM community.

Mathematical Background

A.1 Probabilities and Bayes' Theorem

Joint probability: given two sets of events A and B the joint probability $p(A \cap B)$ is the probability of both occurring. If event A and B are two sets of N continuous outcomes or measurement e.g. $\boldsymbol{y}_A = \{y_1, \ldots, y_N\}$ then assuming for each event the measurements have a joint probability (specifically a PDF) e.g. $p(y_1, \ldots, y_N) = p(\boldsymbol{y}_A)$ the joint between the events can be written $p(\boldsymbol{y}) = p(\boldsymbol{y}_A, \boldsymbol{y}_B)$.

Marginal Probability: given a joint probability $p(\boldsymbol{y}_A, \boldsymbol{y}_B)$ it may be desired to know the probability of a single event. This is performed by integrating out one event from the joint probability i.e. $p(\boldsymbol{y}_A) = \int p(\boldsymbol{y}_A, \boldsymbol{y}_B) d\boldsymbol{y}_B$. This can be performed on probabilities governing multiple events, which can lead to the marginal also being a joint probability. Independence is defined for joint distributions that are composed of a product of marginal distributions for each event, i.e. that each event does not affect the other, the converse is a dependence.

Conditional Probability: given that event B has occurred what is the probability of event A happening $p(A \,|\, B)$. This can also be summarised as the likelihood of event A given event B. A conditional distribution can be calculated via the ratio $p(\boldsymbol{y}_A \,|\, \boldsymbol{y}_B) = p(\boldsymbol{y}_A, \boldsymbol{y}_B)/p(\boldsymbol{y}_B)$ when $p(\boldsymbol{y}_B) > 0$. An independent variable will lead to the marginal and the conditional being equal.

Bayes' Theorem: by simply combing the two conditionals $p(\boldsymbol{y}_A \,|\, \boldsymbol{y}_B)$ and $p(\boldsymbol{y}_B \,|\, \boldsymbol{y}_A)$ (and $p(\boldsymbol{y}_B) > 0$) Bayes theorem is defined:

$$p(\boldsymbol{y}_A \,|\, \boldsymbol{y}_B) = \frac{p(\boldsymbol{y}_B \,|\, \boldsymbol{y}_A) p(\boldsymbol{y}_A)}{p(\boldsymbol{y}_B)} \tag{A.1}$$

In words this can be written as:

$$\text{Posterior} = \frac{\text{Likelihood} \times \text{Prior}}{\text{Evidence}} \tag{A.2}$$

Where the prior is ones initial belief about the event $\boldsymbol{y}_A$ occurring. The likelihood is probability that of observing some outcome $\boldsymbol{y}_B$ given the event $\boldsymbol{y}_A$. The evidence or marginal probability of event $\boldsymbol{y}_B$ occurring is often calculated via integrating out event $\boldsymbol{y}_A$ from the numerator. The posterior is the probability of event $\boldsymbol{y}_A$ given that we have witnessed $\boldsymbol{y}_B$.

Conditional and marginal probabilities therefore sit as key building blocks for any Bayesian analysis.

A.2 Gaussian Identities

Multivariate Gaussian Distribution:

$$\mathcal{N}(\boldsymbol{x} \,|\, \boldsymbol{\mu}, \Sigma) = (2\pi)^{-D/2} |\Sigma|^{-1/2} \exp\left(-\frac{1}{2} (\boldsymbol{x} - \boldsymbol{\mu})^{\mathsf{T}} \Sigma^{-1} (\boldsymbol{x} - \boldsymbol{\mu}) \right) \tag{A.3}$$

Where $\boldsymbol{\mu}$ and Σ are the mean and covariance of size N and $N \times N$.

Conditional of Joint Gaussian Distribution: a joint Gaussian distribution of the random variables $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ is,

$$\begin{bmatrix} \boldsymbol{x}_1 \\ \boldsymbol{x}_2 \end{bmatrix} = \mathcal{N}\left(\begin{bmatrix} \boldsymbol{\mu}_1 \\ \boldsymbol{\mu}_2 \end{bmatrix}, \begin{bmatrix} \Sigma_{1,1} & \Sigma_{1,2} \\ \Sigma_{2,1} & \Sigma_{2,2} \end{bmatrix} \right). \tag{A.4}$$

The conditional distribution is,

$$p\left(\boldsymbol{x_1} \mid \boldsymbol{x_2}\right) \sim \mathcal{N}\left(\boldsymbol{\mu}_1 + \Sigma_{1,2}\Sigma_{2,2}^{-1}(\boldsymbol{x}_2 - \boldsymbol{\mu}_2), \Sigma_{1,1} - \Sigma_{1,2}\Sigma_{2,2}^{-1}\Sigma_{2,1}\right) \tag{A.5}$$

Product of Two Gaussian Distributions: the product of two Gaussian distributions for the same random variable produces another Gaussian distribution with an additional constant,

$$\mathcal{N}\left(\boldsymbol{x} \mid \boldsymbol{\mu_1}, \Sigma_1\right)\mathcal{N}\left(\boldsymbol{x} \mid \boldsymbol{\mu_2}, \Sigma_2\right) = Z\mathcal{N}\left(\boldsymbol{c}, C\right) \tag{A.6}$$

$$\boldsymbol{c} = C(\Sigma_1^{-1}\boldsymbol{\mu}_1 + {\Sigma_2}^{-1}\boldsymbol{\mu}_2) \tag{A.7}$$

$$C = ({\Sigma_1}^{-1} + {\Sigma_2}^{-1})^{-1} \tag{A.8}$$

$$Z = (2\pi)^{-D/2}|\Sigma_1 + \Sigma_2|^{-1/2} \exp\left(-\frac{1}{2}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)^{\mathsf{T}}(\Sigma_1 + \Sigma_2)^{-1}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)\right). \tag{A.9}$$

A.3 Matrix Identities

Woodbury Inversion Lemma:

$$(A + UCV^{\mathsf{T}})^{-1} = A^{-1} + A^{-1}U(C^{-1} + V^{\mathsf{T}}A^{-1}U)^{-1}V^{\mathsf{T}}A^{-1} \tag{A.10}$$

Cholesky Inversion:

The Cholesky decomposition,

$$A = LL^{\mathsf{T}} \tag{A.11}$$

where L is the Cholesky factor (a lower triangular matrix) useful in solving linear systems, i.e. $Ax = b$ where A is positive definite. The approach is numerically stable and the log determinant of A can be calculated using:

$$\log|A| = 2\sum_{i=1}^{N} \log L_{i,i}. \tag{A.12}$$

A.4 Bayesian Calibration and Bias Correction Integrals

Here the closed form solutions to the integrals in stage 2 of BCBC outlined in Section 5.2.2 are formed when:

- The emulator mean function is constant: $H_\eta(\cdot) = 1$
- The emulator covariance function is a separable SE:
  $K_\eta(\cdot,\cdot) = \sigma_\eta^2 \exp\left(-(\boldsymbol{x}-\boldsymbol{x}')^\mathsf{T}\Omega_x(\boldsymbol{x}-\boldsymbol{x}')\right)\exp\left(-(\boldsymbol{t}-\boldsymbol{t}')^\mathsf{T}\Omega_t(\boldsymbol{t}-\boldsymbol{t}')\right)$
- The prior for the parameters is Gaussian: $\boldsymbol{\theta} \sim \mathcal{N}(m_\theta, V_\theta)$

The mean function integrals are:

a)

$$\int H_\eta(D_{z,i}(\boldsymbol{\theta}))p(\boldsymbol{\theta})\mathrm{d}\boldsymbol{\theta} = 1\int p(\boldsymbol{\theta})\mathrm{d}\boldsymbol{\theta} = 1 \tag{A.13}$$

b)

$$\int K_\eta(D_{z,i}(\boldsymbol{\theta}), D_{y,j})^\mathsf{T} p(\boldsymbol{\theta})\mathrm{d}\boldsymbol{\theta} = \int \sigma_\eta^2 \exp\left(-(x_i^z - x_j)^\mathsf{T}\Omega_x(x_i^z - x_j)\right) \exp\left(-(\boldsymbol{\theta}-t_j)^\mathsf{T}\Omega_t(\boldsymbol{\theta}-t_j)\right)\mathcal{N}(m_\theta, V_\theta)\mathrm{d}\boldsymbol{\theta} \tag{A.14}$$

Part of the covariance function is not dependent on $\boldsymbol{\theta}$ and is constant resulting in Eqs. (A.15) and (A.16).

$$C_k = \sigma_\eta^2 \exp\left(-(x_i^z - x_j)^\mathsf{T}\Omega_x(x_i^z - x_j)\right) \tag{A.15}$$

$$\int K_\eta \left(D_{z,i}(\boldsymbol{\theta}), D_{y,j}\right)^\mathsf{T} p\left(\boldsymbol{\theta}\right) \mathrm{d}\boldsymbol{\theta} = C_k \int \exp\left(-(\boldsymbol{\theta} - t_j)^\mathsf{T} \Omega_t (\boldsymbol{\theta} - t_j)\right) \mathcal{N}\left(m_\theta, V_\theta\right) \mathrm{d}\boldsymbol{\theta} \tag{A.16}$$

The expression of the covariance function dependent on $\boldsymbol{\theta}$ can be expressed as a Gaussian distribution Eq. (A.17).

$$\frac{(2\pi)^{n/2}|(2\Omega_t)^{-1}|^{-1/2}}{(2\pi)^{n/2}|(2\Omega_t)^{-1}|^{-1/2}} \exp\left(-\frac{1}{2} - (\boldsymbol{\theta} - t_j)^\mathsf{T} (2\Omega_t)(\boldsymbol{\theta} - t_j)\right) = \frac{1}{(2\pi)^{n/2}|(2\Omega_t)^{-1}|^{-1/2}} \mathcal{N}\left(t_j, (2\Omega_t)^{-1}\right) \tag{A.17}$$

As a result the integral is the product of two Gaussian distribution multiplied by a constant.

$$\int K_\eta \left(D_{z,i}(\boldsymbol{\theta}), D_{y,j}\right)^\mathsf{T} p\left(\boldsymbol{\theta}\right) \mathrm{d}\boldsymbol{\theta} = \frac{C_k}{(2\pi)^{n/2}|(2\Omega_t)^{-1}|^{-1/2}} \int \mathcal{N}\left(t_j, (2\Omega_t)^{-1}\right) \mathcal{N}\left(m_\theta, V_\theta\right) \mathrm{d}\boldsymbol{\theta} \tag{A.18}$$

The product of two Gaussian distributions is an unnormalised Gaussian distribution as shown in Appendix A.2 i.e. $\mathcal{N}\left(t_j, (2\Omega_t)^{-1}\right) \mathcal{N}\left(m_\theta, V_\theta\right) = Z \mathcal{N}\left(\boldsymbol{\mu}, \Sigma\right)$. The normalising constant Z is shown in Eq. (A.19).

$$Z = (2\pi)^{-n/2} |V_\theta + (2\Omega_t)^{-1}|^{-1/2} \exp\left(-\frac{1}{2}(m_\theta - t_j)^\mathsf{T} (V_\theta + (2\Omega_t)^{-1})^{-1} (m_\theta - t_j)\right) \tag{A.19}$$

As the integral $\int \mathcal{N}\left(\boldsymbol{\mu}, \Sigma\right) \mathrm{d}\boldsymbol{\theta} = 1$, meaning that the marginalisation integral is equal to the product of the constants.

$$\int K_\eta \left(D_{z,i}(\boldsymbol{\theta}), D_{y,j}\right)^\mathsf{T} p\left(\boldsymbol{\theta}\right) \mathrm{d}\boldsymbol{\theta} = \frac{C_k Z}{(2\pi)^{n/2}|(2\Omega_t)^{-1}|^{-1/2}} \tag{A.20}$$

Equation (A.20) simplifies to Eq. (A.21).

$$\sigma_\eta^2|\mathbb{I}+2V_\theta\Omega_t|^{-1/2}\exp\left(-(x_i^z-x_j)^\mathsf{T}\Omega_x(x_i^z-x_j)\right) \\ \exp\left(-(m_\theta-t_j)^\mathsf{T}(V_\theta+(2\Omega_t)^{-1})^{-1}(m_\theta-t_j)\right) \quad \text{(A.21)}$$

The covariance function integrals are:

c)

$$\int K_\eta\left(D_{z,i}(\boldsymbol{\theta}),D_{z,j}(\boldsymbol{\theta})\right)p\left(\boldsymbol{\theta}\right)\mathrm{d}\boldsymbol{\theta}=\int\sigma_\eta^2\exp\left(-(x_i^z-x_j)^\mathsf{T}\Omega_x(x_i^z-x_j)\right) \\ \exp\left(-(\boldsymbol{\theta}-\boldsymbol{\theta})^\mathsf{T}\Omega_t(\boldsymbol{\theta}-\boldsymbol{\theta})\right)\mathcal{N}\left(m_\theta,V_\theta\right)\mathrm{d}\boldsymbol{\theta} \quad \text{(A.22)}$$

The only part of the covariance function dependent on $\boldsymbol{\theta}$ contains $(\boldsymbol{\theta}-\boldsymbol{\theta})=0$, leading to Eqs. (A.23) and (A.24).

$$\int K_\eta\left(D_{z,i}(\boldsymbol{\theta}),D_{z,j}(\boldsymbol{\theta})\right)p\left(\boldsymbol{\theta}\right)\mathrm{d}\boldsymbol{\theta}=\sigma_\eta^2\exp\left(-(x_i^z-x_j)^\mathsf{T}\Omega_x(x_i^z-x_j)\right)\int\mathcal{N}\left(m_\theta,V_\theta\right)\mathrm{d}\boldsymbol{\theta} \quad \text{(A.23)}$$

$$\int K_\eta\left(D_{z,i}(\boldsymbol{\theta}),D_{z,j}(\boldsymbol{\theta})\right)p\left(\boldsymbol{\theta}\right)\mathrm{d}\boldsymbol{\theta}=\sigma_\eta^2\exp\left(-(x_i^z-x_j)^\mathsf{T}\Omega_x(x_i^z-x_j)\right) \quad \text{(A.24)}$$

d)

$$\int K_\eta\left(D_{z,j}(\boldsymbol{\theta}),D_{y,k}\right)K_\eta\left(D_{z,i}(\boldsymbol{\theta}),D_{y,l}\right)^\mathsf{T}p\left(\boldsymbol{\theta}\right)\mathrm{d}\boldsymbol{\theta}= \\ \int\sigma_\eta^2\exp\left(-(x_j^z-x_k)^\mathsf{T}\Omega_x(x_j^z-x_k)\right)\exp\left(-(\boldsymbol{\theta}-t_k)^\mathsf{T}\Omega_t(\boldsymbol{\theta}-t_k)\right) \\ \sigma_\eta^2\exp\left(-(x_i^z-x_l)^\mathsf{T}\Omega_x(x_i^z-x_l)\right)\exp\left(-(\boldsymbol{\theta}-t_l)^\mathsf{T}\Omega_t(\boldsymbol{\theta}-t_l)\right)\mathcal{N}\left(m_\theta,V_\theta\right)\mathrm{d}\boldsymbol{\theta} \quad \text{(A.25)}$$

Again by collecting the constants in Eq. (A.26) and forming Gaussian distributions from the remaining two covariance functions dependent on $\boldsymbol{\theta}$, the problem becomes the integral of the product of three Gaussian distributions in Eq. (A.27).

$$C_k = \sigma_\eta^4 \exp\left(-(x_j^z - x_k)^\mathsf{T}\Omega_x(x_j^z - x_k)\right) \exp\left(-(x_i^z - x_l)^\mathsf{T}\Omega_x(x_i^z - x_l)\right) \quad \text{(A.26)}$$

$$\int K_\eta\left(D_{z,j}(\boldsymbol{\theta}), D_{y,k}\right) K_\eta\left(D_{z,i}(\boldsymbol{\theta}), D_{y,l}\right)^\mathsf{T} p\left(\boldsymbol{\theta}\right)\mathrm{d}\boldsymbol{\theta} = \frac{C_k}{((2\pi)^{n/2}|(2\Omega_t)^{-1}|^{-1/2})^2} \int \mathcal{N}\left(t_k, (2\Omega_t)^{-1}\right)\mathcal{N}\left(t_l, (2\Omega_t)^{-1}\right)\mathcal{N}\left(m_\theta, V_\theta\right)\mathrm{d}\boldsymbol{\theta} \quad \text{(A.27)}$$

The first product (using Appendix A.2) is $\mathcal{N}\left(t_k, (2\Omega_t)^{-1}\right)\mathcal{N}\left(t_l, (2\Omega_t)^{-1}\right) = Z_1$ $\mathcal{N}\left((t_k + t_l)/2, \Omega_t^{-1}/4\right)$ and the second product $Z_1\mathcal{N}\left((t_k + t_l)/2, \Omega_t^{-1}/4\right)\mathcal{N}\left(m_\theta, V_\theta\right) = Z_1 Z_2 \mathcal{N}\left(\boldsymbol{\mu}, \Sigma\right)$. As this single Gaussian distribution integrates to one the marginalisation integral is the product of the constants shown in Eq. (A.28).

$$\int K_\eta\left(D_{z,j}(\boldsymbol{\theta}), D_{y,k}\right) K_\eta\left(D_{z,i}(\boldsymbol{\theta}), D_{y,l}\right)^\mathsf{T} p\left(\boldsymbol{\theta}\right)\mathrm{d}\boldsymbol{\theta} = \frac{C_k Z_1 Z_2}{((2\pi)^{n/2}|(2\Omega_t)^{-1}|^{-1/2})^2} \quad \text{(A.28)}$$

Equation (A.28) simplifies to Equation (A.29).

$$\sigma_\eta^4 |\mathbb{I} + 4V_\theta\Omega_t|^{-1/2} \exp\left(-(x_j^z - x_k)^\mathsf{T}\Omega_x(x_j^z - x_k) - (x_i^z - x_l)^\mathsf{T}\Omega_x(x_i^z - x_l)\right) \exp\left(-\frac{1}{2}(t_k - t_l)^\mathsf{T}\Omega_t(t_k - t_l) - \frac{1}{2}\left(m_\theta - \frac{t_k + t_l}{2}\right)^\mathsf{T}\left(V_\theta + \frac{\Omega_t^{-1}}{4}\right)^{-1}\left(m_\theta - \frac{t_k + t_l}{2}\right)\right) \quad \text{(A.29)}$$

e)

$$\int H_\eta\left(D_{z,j}(\boldsymbol{\theta})\right) H_\eta\left(D_{z,i}(\boldsymbol{\theta})\right)^\mathsf{T} p\left(\boldsymbol{\theta}\right)\mathrm{d}\boldsymbol{\theta} = 1 \quad \text{(A.30)}$$

f)

$$\int K_\eta\left(D_{z,j}(\boldsymbol{\theta}), D_{y,k}\right) H_\eta\left(D_{z,i}(\boldsymbol{\theta})\right)^\mathsf{T} p\left(\boldsymbol{\theta}\right)\mathrm{d}\boldsymbol{\theta} = \sigma_\eta^2|\mathbb{I} + 2V_\theta\Omega_t|^{-1/2}$$
$$\exp\left(-(x_j^z - x_k)^\mathsf{T}\Omega_x(x_j^z - x_k)\right)\exp\left(-\frac{1}{2}(m_\theta - t_k)^\mathsf{T}(V_\theta + (2\Omega_t)^{-1})^{-1}(m_\theta - t_k)\right)$$
$$\mathbb{E}_\theta\left(H_\eta\left(D_{z,i}(\boldsymbol{\theta})\right)^\mathsf{T}\right) \quad \text{(A.31)}$$

This results from the same process as b) however includes the expectation of the design matrix with respect to $\boldsymbol{\theta}$, $\mathbb{E}_\theta\left(H_\eta\left(D_{z,i}(\boldsymbol{\theta})\right)^\mathsf{T}\right)$. When a constant mean function is implemented this term equals one.

g)

$$\int H_\eta\left(D_{z,j}(\boldsymbol{\theta})\right) K_\eta\left(D_{z,i}(\boldsymbol{\theta}), D_{y,l}\right)^\mathsf{T} p\left(\boldsymbol{\theta}\right)\mathrm{d}\boldsymbol{\theta} = \sigma_\eta^2|\mathbb{I} + 2V_\theta\Omega_t|^{-1/2}$$
$$\exp\left(-(x_i^z - x_l)^\mathsf{T}\Omega_x(x_i^z - x_l)\right)\exp\left(-\frac{1}{2}(m_\theta - t_l)^\mathsf{T}(V_\theta + (2\Omega_t)^{-1})^{-1}(m_\theta - t_l)\right)$$
$$\mathbb{E}_\theta\left(H_\eta\left(D_{z,j}(\boldsymbol{\theta})\right)\right) \quad \text{(A.32)}$$

A.5 Golub-Welsch Algorithm

The weights and nodes for the Gauss-Hermite quadrature can be calculated using the Golub-Welsch algorithm [148]. This method uses the recurrence relationship of a set of orthogonal polynomials $(\phi_i(x))$ shown in Eq. (A.33).

$$\phi_j(x) = (a_j x + b_j)\phi_{j-1}(x) - c_j\phi_{j-2}(x) \quad \text{(A.33)}$$

Where $j = 1, \cdots, N-1$; $\phi_{-1}(x) = 0$ and $\phi_0(x) = 1$. This form can be used to construct the matrix equation in Eq. (A.34).

$$x\boldsymbol{\phi}(x) = T\boldsymbol{\phi}(x) + \phi_N(x)/a_N\boldsymbol{e} \quad \text{(A.34)}$$

Where $\boldsymbol{\phi}$ is the vector $[\phi_0(x), ..., \phi_{N-1}(x)]^\mathsf{T}$, T is a matrix of the coefficients a_j, b_j and c_j and $\boldsymbol{e} = [0, \cdots, 1]^\mathsf{T}$. By performing a diagonal similarity transformation the

tridiagonal matrix J is constructed.

$$\mathbf{DTD}^{-1} = \mathbf{J} = \begin{bmatrix} \alpha_1 & \beta_1 & 0 & & & \\ \beta_1 & \alpha_2 & \beta_2 & & & \\ 0 & \beta_2 & \alpha_3 & & & \\ & & & \ddots & & \\ & & & & \alpha_{N-1} & \beta_{N-1} \\ & & & & \beta_{N-1} & \alpha_N \end{bmatrix} \tag{A.35}$$

Where the coefficients are, $\alpha_j = -\frac{b_j}{a_j}$ and $\beta_j = \sqrt{\frac{c_{j+1}}{a_j a_{j+1}}}$.

This leads to the matrix equation,

$$J\boldsymbol{\phi}(x_i) = x_i\boldsymbol{\phi}(x_i), \tag{A.36}$$

where x_i are the nodes of the Gaussian quadrature which are the eigenvalues of J. The weights w_i can be determined from the eigenvectors of J as in Eq. (A.37).

$$w_i = \beta_0 \frac{v_i^2}{||v_i||_2^2} \tag{A.37}$$

Where,

$$\beta_0 = \int_b^a w(x)dx. \tag{A.38}$$

Here $w(x)$ is the weight function of the Gaussian quadrature, a and b the Gaussian quadrature limits and v_i is the ith eigenvector.

Publications

B.1 Journal Papers

R. Fuentes, P. A. Gardner, C. Mineo, T. J. Rogers, S. G. Pierece, K. Worden, N. Dervilis, E. J. Cross, 2018, Autonomous Ultrasonic Inspection using Bayesian Optimisation and Robust Outlier Analysis. Submitted to Mechanical Systems and Signal Processing.

B.2 Reviewed Conference Papers

P. Gardner, T. J. Rogers, C. Lord and R. J. Barthorpe, 2018, Sparse Gaussian Process Emulators for Surrogate Design Modelling, Proceedings of the 3rd International Conference on Uncertainty in Mechanical Engineering, Darmstadt, Germany. In press.

P. Gardner, C. Lord, R. J. Barthorpe, 2018, An Evaluation of Validation Metrics for Probabilistic Model Outputs, Proceedings of the ASME 2018 Verification and Validation Symposium, Minneapolis, USA.

B.3 Conference Papers

P. Gardner, C. Lord, R. J. Barthorpe, 2018, A Multi-Level Uncertainty Integration Strategy for Forward Model-Driven SHM, Proceedings of ISMA2018 International Conference on Noise and Vibration Engineering, Leuven, Belgium.

P. Gardner, C. Lord, R. J. Barthorpe, 2018, A Probabilistic Framework for Forward Model-Driven SHM, Proceedings of the 9th European Workshop on Structural Health Monitoring, Manchester, UK.

P. Gardner, C. Lord, R. J. Barthorpe, 2018, Bayesian History Matching for Forward Model-Driven Structural Health Monitoring, Proceedings of IMAC XXXVI International Conference on Modal Analysis, Florida, USA.

P. Gardner, R. J. Barthorpe, C. Lord, 2017, Bayesian Calibration and Bias Correction for Forward Model-Driven SHM, Proceedings of the 11th International Workshop on Structural Health Monitoring, Stanford, USA.

P. Gardner, R. J. Barthorpe, C. Lord, 2016, The Development of a Damage Model for the use in Machine Learning Driven SHM and Comparison with Conventional SHM Methods, Proceedings of ISMA2016 International Conference on Noise and Vibration Engineering, Leuven, Belgium.

P. Gardner, R. J. Barthorpe, C. Lord, 2016, Quantification of Uncertainty for Experimentally Obtained Modal Parameters in the Creation of a Robust Damage Model, Proceedings of the 6th European Conference on Structural Control, Sheffield, UK.

Bibliography

[1] C. R. Farrar and K. Worden. An introduction to structural health monitoring. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 365(1851):303–315, 2007.

[2] A. Rytter. *Vibrational Based Inspection of Civil Engineering Structures*. PhD thesis, 1993.

[3] C. R. Farrar and N. A. Lieven. Damage prognosis: The future of structural health monitoring. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 365:623–632, 2007.

[4] C. R. Farrar and K. Worden. *Structural Health Monitoring: A Machine Learning Perspective*. 2013.

[5] K. Worden and G. Manson. The application of machine learning to Structural Health Monitoring. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 365(1851):515–537, 2007.

[6] W. Fan and P. Qiao. Vibration-based damage identification methods: A review and comparative study. *Structural Health Monitoring*, 10(1):83–111, 2011.

[7] M. Vagnoli, R. Remenyte-Prescott, and J. Andrews. Railway bridge structural health monitoring and fault detection: State-of-the-art methods and future challenges. *Structural Health Monitoring*, 17(4):971–1007, 2018.

[8] M. I. Friswell and J. E. Mottershead. Inverse methods in structural health monitoring. *Key Engineering Materials*, 204-205:201–210, 2001.

[9] M. I. Friswell. Damage identification using inverse methods. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 365(1851):393–410, feb 2007.

[10] H. Sohn, C. R. Farrar, N. F. Hunter, and K. Worden. Structural Health Monitoring Using Statistical Pattern Recognition Techniques. *Journal of Dynamic Systems, Measurement, and Control*, 123(4):706, 2001.

[11] R. J. Barthorpe. *On Model- and Data-based Approaches to Structural Health Monitoring*. PhD thesis, University of Sheffield, 2011.

[12] M. I. Friswell and J. E. Mottershead. *Finite Element Model Updating in Structural Dynamics*. 1995.

[13] M. I. Friswell and J. E. Mottershead. Physical understanding of structures by model updating. In *Proceedings of International Conference on Structural System Identification*, pages 81–96, 2001.

[14] J. E. Mottershead, M. Link, and M. I. Friswell. The sensitivity method in finite element model updating: A tutorial. *Mechanical Systems and Signal Processing*, 2011.

[15] B. Jaishi and W.-X. Ren. Structural finite element model updating using ambient vibration test results. *Journal of Structural Engineering*, 2005.

[16] E. Simoen, G. De Roeck, and G. Lombaert. Dealing with uncertainty in model updating for damage assessment: A review. *Mechanical Systems and Signal Processing*, 56:123–149, 2015.

[17] T. Haag, S. Carvajal González, and M. Hanss. Model validation and selection based on inverse fuzzy arithmetic. *Mechanical Systems and Signal Processing*, 32:116–134, 2012.

[18] C. Rasmussen and Z. Ghahramani. Occam's razor. *Advances in neural information processing systems 13: proceedings of the 2000 conference*, page 294, 2001.

[19] J. L. Beck and L. S. Katafygiotis. Updating models and their uncertainties. I: Bayesian statistical framework. *Journal of Engineering Mechanics*, 124(4): 455–461, 1998.

[20] L. S. Katafygiotis and J. L. Beck. Updating models and their uncertainties. II: model identifiability. *Journal of Engineering Mechanics*, 124(4):463–467, 1998.

[21] M. I. Friswell, J. E. T. Penny, and S. D. Garvey. Parameter subset selection in damage location. *Inverse Problems in Engineering*, 5(3):189–215, 1997.

[22] M. I. Friswell, J. E. Mottershead, and H. Ahmadian. Combining subset selection and parameter constraints in model updating. *Journal of Vibration and Acoustics, Transactions of the ASME*, 120(OCTOBER):854–859, 1998.

[23] V. Cherkassky and F. Mulier. *Learning from Data: Concepts, Theory, and Methods*. 1998.

[24] C. M. Bishop. *Pattern Recognition and Machine Learning*. 2006.

[25] M. Tipping. Sparse Bayesian Learning and the Relevance Vector Mach. *Journal of Machine Learning Research*, 1:211–244, 2001.

[26] C. M. Bishop. Neural Networks for Pattern Recognition. *Journal of the American Statistical Association*, 1995.

[27] C. E. Rasmussen and C. K. I. Williams. *Gaussian processes for machine learning*. MIT Press, 2006.

[28] R. J. E. Fuentes. *On Bayesian Networks for Structural Health and Condition Monitoring*. PhD thesis, University of Sheffield, 2017.

[29] J. M. Ko, Z. G. Sun, and Y. Q. Ni. Multi-stage identification scheme for detecting damage in cable-stayed Kap Shui Mun Bridge. *Engineering Structures*, 2002.

[30] J. J. Lee, J. W. Lee, J. H. Yi, C. B. Yun, and H. Y. Jung. Neural networks-based damage detection for bridges considering errors in baseline finite element models. *Journal of Sound and Vibration*, 280(3-5):555–578, 2005.

[31] S. B. Satpal, A. Guha, and S. Banerjee. Damage identification in aluminum beams using support vector machine: Numerical and experimental studies. *Structural Control and Health Monitoring*, 2016.

[32] M. A. Hariri-Ardebili and F. Pourkamali-Anaraki. Support vector machine based reliability analysis of concrete dams. *Soil Dynamics and Earthquake Engineering*, 104(September 2017):276–295, 2018.

[33] D. J. Spiegelhalter, N. G. Best, B. P. Carlin, and A. Van Der Linde. Bayesian measures of model complexity and fit. *Journal of the Royal Statistical Society. Series B: Statistical Methodology*, 64(4):583–616, 2002.

[34] J. E. Oakley and A. O'Hagan. Probabilistic sensitivity analysis of complex models: A Bayesian approach. *Journal of the Royal Statistical Society. Series B: Statistical Methodology*, 66(3):751–769, 2004.

[35] B. Sudret. Global sensitivity analysis using polynomial chaos expansions. *Reliability Engineering and System Safety*, 93(7):964–979, 2008.

[36] N. Fajraoui, F. Ramasomanana, A. Younes, T. A. Mara, P. Ackerer, and A. Guadagnini. Use of global sensitivity analysis and polynomial chaos expansion for interpretation of nonreactive transport experiments in laboratory-scale porous media. *Water Resources Research*, 47(2):1–14, 2011.

[37] M. Meo and G. Zumpano. On the optimal sensor placement techniques for a bridge structure. *Engineering Structures*, 27(10):1488–1497, 2005.

[38] W. Liu, W. cheng Gao, Y. Sun, and M. jian Xu. Optimal sensor placement for spatial lattice structure based on genetic algorithms. *Journal of Sound and Vibration*, 317(1-2):175–189, 2008.

[39] E. B. Flynn and M. D. Todd. A Bayesian approach to optimal sensor placement for structural health monitoring with application to active sensing. *Mechanical Systems and Signal Processing*, 24(4):891–903, 2010.

[40] W. L. Oberkampf and C. J. Roy. *Verification and Validation in Scientific Computing*. Cambridge University Press, Cambridge, 2010.

[41] G. E. P. Box and N. R. Draper. *Emperical model-building and response surfaces*. 1987.

[42] Standard for Verification and Validation in Computational Fluid Dynamics and Heat Transfer: ASME V&V 20, 2009.

[43] F. J. Massey. The Kolmogorov-Smirnov test for goodness of fit. *Journal of the American Statistical Association*, 1951.

[44] L. H. Miller. Table of percentage points of Kolmogorov statistics. *Journal of the American Statistical Association*, 1956.

[45] A. Gretton. A kernel two-Sample test. *Journal of Machine Learning Research*, 13:723–773, 2012.

[46] A. Gretton, K. Fukumizu, C. H. Teo, L. Song, B. Schölkopf, and A. J. Smola. A kernel statistical test of independence. *Neural Information Processing Systems*, pages 585–592, 2008.

[47] A. Gretton, B. Sriperumbudur, D. Sejdinovic, H. Strathmann, and M. Pontil. Optimal kernel choice for large-scale two-sample tests. *Advances in Neural Information Processing Systems 25*, pages 1214–1222, 2012.

[48] A. G. Dougal J. Sutherland, Hsiao-Yu Tung, Heiko Strathmann, Soumyajit De, Aaditya Ramdas, Alex Smola. Generative models and model criticism via optimized maximum mean discrepancy. *Proceedings of ICLR 2017*, apr 2017.

[49] J. R. Lloyd and Z. Ghahramani. Statistical Model Criticism using Kernel Two Sample Tests. In *Advances in Neural Information Processing Systems*, 2015.

[50] K. Chwialkowski, H. Strathmann, and A. Gretton. A kernel test of goodness of fit. In *Proceedings of the 33rd International Conference on Machine Learning*, volume 48, 2016.

[51] W. Jitkrittum, W. Xu, Z. Szabo, K. Fukumizu, and A. Gretton. A linear-time kernel goodness-of-fit test. In *Advances in Neural Information Processing Systems*, 2017.

[52] S. Sankararaman and S. Mahadevan. Integration of model verification, validation, and calibration for uncertainty quantification in engineering systems. *Reliability Engineering & System Safety*, 138:194–209, 2015.

[53] M. Lavine and M. J. Schervish. Bayes factors: what they are and what they are not. *The American Statistician*, 1999.

[54] S. Sankararaman and S. Mahadevan. Assessing the reliability of computational models under uncertainty. In *54th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, Structures, Structural Dynamics, and Materials and Co-located Conferences. American Institute of Aeronautics and Astronautics, 2013.

[55] C. Li and S. Mahadevan. Role of calibration, validation, and relevance in multi-level uncertainty integration. *Reliability Engineering and System Safety*, 148:32–43, 2016.

[56] D. M. Blei, A. Kucukelbir, and J. D. McAuliffe. Variational Inference: A Review for Statisticians. *Journal of the American Statistical Association*, 112 (518):859–877, 2017.

[57] K. P. Murphy. *Machine Learning: A Probabilistic Perspective*. 2012.

[58] J. Lin. Divergence measures based on the Shannon entropy. *IEEE Transactions on Information Theory*, 37(1):145–151, 1991.

[59] Q. Wang, S. R. Kulkarni, and S. Verdú. Divergence estimation of continuous distributions based on data-dependent partitions. *IEEE Transactions on Information Theory*, 51(9):3064–3074, 2005.

[60] X. L. Nguyen, M. J. Wainwright, and M. I. Jordan. Nonparametric estimation of the likelihood ratio and divergence functionals. In *IEEE International Symposium on Information Theory - Proceedings*, pages 2016–2020, 2007.

[61] B. K. Sriperumbudur, K. Fukumizu, A. Gretton, B. Schölkopf, and G. R. G. Lanckriet. On integral probability metrics, phi-divergences and binary classification. (1):1–18, 2009.

[62] S. Ferson, W. L. Oberkampf, and L. Ginzburg. Model validation and predictive capability for the thermal challenge problem. *Computer Methods in Applied Mechanics and Engineering*, 197(29-32):2408–2430, 2008.

[63] Y. Liu, W. Chen, P. Arendt, and H.-Z. Huang. Toward a better understanding of model validation metrics. *Journal of Mechanical Design*, 133(7):071005, 2011.

[64] H. Xu, Z. Jiang, D. W. Apley, and W. Chen. New metrics for validation of data-driven random process models in uncertainty quantification. *Journal of Verification, Validation and Uncertainty Quantification*, 1(2):021002, 2015.

[65] Z. Wang, Y. Fu, R.-J. Yang, S. Barbat, and W. Chen. Validating dynamic engineering models under uncertainty. *Journal of Mechanical Design*, 138(11): 111402, 2016.

[66] C. C. Aggarwal, A. Hinneburg, and D. A. Keim. On the surprising behavior of distance metrics in high dimensional space. *International Conference on Database Theory*, pages 420–434, 2001.

[67] P. S. Craig, M. Goldstein, A. H. Seheult, and J. A. Smith. Pressure matching for hydrocarbon reservoirs: a case study in the use of Bayes linear strategies for large computer experiments. In *Lecture Notes in Statistics*, pages 37–93. 1997.

[68] M. C. Kennedy and A. O'Hagan. Bayesian calibration of computer models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 63 (3):425–464, 2001.

[69] M. J. Bayarri, J. O. Berger, R. Paulo, J. Sacks, J. A. Cafeo, J. Cavendish, C.-H. Lin, and J. Tu. A framework for validation of computer models. *Technometrics*, 49(2):138–154, 2007.

[70] M. Goldstein, L. House, and J. Rougier. Assessing model discrepancy using a multi-model ensemble. *Sciences-New York*, pages 1–35, 2008.

[71] I. Andrianakis, I. R. Vernon, N. McCreesh, T. J. McKinley, J. E. Oakley, R. N. Nsubuga, M. Goldstein, and R. G. White. Bayesian history matching of complex infectious disease models using emulation: a tutorial and a case study on HIV in Uganda. *PLoS Computational Biology*, 11(1), 2015.

[72] R. Tripathy and I. Bilionis. Deep UQ: Learning deep neural network surrogate models for high dimensional uncertainty quantification, 2018.

[73] Y. Jin. A comprehensive survey of fitness approximation in evolutionary computation. *Soft Computing*, 9(1):3–12, 2005.

[74] T. Mengistu and W. Ghaly. Aerodynamic optimization of turbomachinery blades using evolutionary methods and ANN-based surrogate models. *Optimization and Engineering*, 9(3):239–255, 2008.

[75] Z. Jiang and M. Gu. Optimization of a fender structure for the crashworthiness design. *Materials and Design*, 31(3):1085–1095, 2010.

[76] S. Razavi, B. A. Tolson, and D. H. Burn. Numerical assessment of metamodelling strategies in computationally intensive optimization. *Environmental Modelling and Software*, 34:67–86, 2012.

[77] S. Chen, Z. Jiang, S. Yang, and W. Chen. Multimodel fusion based sequential optimization. *AIAA Journal*, 55(1):241–254, 2017.

[78] J. Sacks, W. J. Welch, T. J. Mitchell, and H. P. Wynn. Design and analysis of computer experiments. *Statistical Science*, 4(4):409–423, 1989.

[79] J. Sacks, S. Schiller, and W. Welch. Designs for computer experiments. *Technometrics*, 31(1):41–47, 1989.

[80] C. M. Bishop. Training with noise is equivalent to Tikhonov regularization. *Neural Computation*, 7(1):108–116, 1995.

[81] D. R. Broad. Water distribution system optimization using metamodels. *Journal of Water Resources Planning and Management*, 131(3):172–180, 2005.

[82] J. Sreekanth and B. Datta. Multi-objective management of saltwater intrusion in coastal aquifers using genetic programming and modular neural network based surrogate models. *Journal of Hydrology*, 393(3-4):245–256, 2010.

[83] Z. Zhang, K. Shankar, T. Ray, E. V. Morozov, and M. Tahtali. Vibration-based inverse algorithms for detection of delamination in composites. *Composite Structures*, 102:226–236, 2013.

[84] K. O. Stanley and R. Miikkulainen. Evolving neural networks through augmenting topologies. *Evolutionary Computation*, 10(2):99–127, 2002.

[85] N. E. Owen, P. Challenor, P. P. Menon, and S. Bennani. Comparison of surrogate-based uncertainty quantification methods for computationally expensive simulators. *SIAM/ASA Journal on Uncertainty Quantification*, 5(1): 403–435, 2017.

[86] S. Hosder, R. Walters, and M. Balch. Efficient sampling for non-intrusive polynomial chaos spplications with multiple uncertain input variables. *48th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, pages 1–16, 2007.

[87] A. O'Hagan. Polynomial chaos: a tutorial and critique from a statisticians perspective. *SIAM/ASA Journal on Uncertainty Quantification*, 2013.

[88] H. N. Najm. Uncertainty quantification and polynomial chaos techniques in computational fluid dynamics. *Annual Review of Fluid Mechanics*, 41(1):35–52, 2009.

[89] J. Li and D. Xiu. Evaluation of failure probability via surrogate models. *Journal of Computational Physics*, 229(23):8966–8980, 2010.

[90] E. Laloy, B. Rogiers, J. A. Vrugt, D. Mallants, and D. Jacques. Efficient posterior exploration of a high-dimensional groundwater model from two-stage Markov chain Monte Carlo simulation and polynomial chaos expansion. *Water Resources Research*, 49(5):2664–2682, 2013.

[91] Y. J. Kim. Comparative study of surrogate models for uncertainty quantification of building energy model: Gaussian process emulator vs. polynomial chaos expansion. *Energy and Buildings*, 133:46–58, 2016.

[92] K. Konakli and B. Sudret. Polynomial meta-models with canonical low-rank approximations: Numerical insights and comparison to sparse polynomial chaos expansions. *Journal of Computational Physics*, 321:1144–1169, 2016.

[93] M. Goldstein and D. Wooff. *Bayes linear statistics: Theory and methods*. 2007.

[94] I. Vernon, M. Goldsteiny, and R. G. Bowerz. Galaxy formation: a Bayesian uncertainty analysis. *Bayesian Analysis*, 5(4):619–670, 2010.

[95] P. S. Craig, M. Goldstein, J. C. Rougier, and A. H. Seheult. Bayesian forecasting for complex systems using computer simulations. *Journal of the American Statistical Association*, 96(454):717–729, 2001.

[96] M. Goldstein and R. Paulo. External Bayesian analysis for computer simulators. *Bayesian Statistics 9*, (1996), 2012.

[97] M. Jones, M. Goldstein, P. Jonathan, and D. Randell. Bayes linear analysis for Bayesian optimal experimental design. *Journal of Statistical Planning and Inference*, pages 115–129, 2015.

[98] M. Jones, M. Goldstein, P. Jonathan, and D. Randell. Bayes linear analysis for sequential optimal design, 2016.

[99] A. O'Hagan and J. F. C. Kingman. Curve fitting and optimal design for prediction. *Journal of the Royal Statistical Society. Series B (Methodological)*, 40(1):1–42, 1978.

[100] N. Cressie. *Statistics for Spatial Data*. 1992.

[101] A. O'Hagan. Bayesian analysis of computer code outputs: A tutorial. *Reliability Engineering & System Safety*, 91(10-11):1290–1300, oct 2006.

[102] Y. Ling, J. Mullins, and S. Mahadevan. Calibration of multi-physics computational models using Bayesian networks. pages 1–38, 2012.

[103] D. Higdon, J. Gattiker, B. Williams, and M. Rightley. Computer model calibration using high-dimensional output. *Journal of the American Statistical Association*, 103(482):570–583, 2008.

[104] S. Conti and A. O'Hagan. Bayesian emulation of complex multi-output and dynamic computer models. *Journal of Statistical Planning and Inference*, 140 (3):640–651, 2010.

[105] P. D. Arendt, D. W. Apley, and W. Chen. Quantification of model uncertainty: calibration, model discrepancy, and identifiability. *Journal of Mechanical Design*, 134(10):100908, 2012.

[106] Z. Jiang, S. Chen, D. W. Apley, and W. Chen. Reduction of epistemic model uncertainty in simulation-based multidisciplinary design. *Journal of Mechanical Design*, 138(8):081403, 2016.

[107] T. E. Fricker, J. E. Oakley, and N. M. Urban. Multivariate Gaussian process emulators with nonseparable covariance structures. *Technometrics*, 55(1):47–56, 2013.

[108] P. B. Holden, N. R. Edwards, J. Hensman, and R. D. Wilkinson. ABC for climate: dealing with expensive simulators. *Handbook of ABC*, pages 1–28, 2015.

[109] I. Andrianakis, I. Vernon, N. McCreesh, T. J. McKinley, J. E. Oakley, R. N. Nsubuga, M. Goldstein, and R. G. White. History matching of a complex epidemiological model of human immunodeficiency virus transmission by using variance emulation. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 66(4):717–740, aug 2017.

[110] J. E. Oakley and B. D. Youngman. Calibration of stochastic computer simulators using likelihood emulation. *Technometrics*, 59(1):80–92, jan 2017.

[111] R. B. Gramacy and H. K. Lee. Bayesian treed Gaussian process models with an application to computer modeling. *Journal of the American Statistical Association*, 103(483):1119–1130, 2008.

[112] R. Calandra, J. Peters, C. E. Rasmussen, and M. P. Deisenroth. Manifold Gaussian processes for regression. In *Advances in neural information processing systems*, 2016.

[113] R. M. Neal. *Bayesian Learning for Neural Networks*. PhD thesis, 1995.

[114] J. Oakley. Eliciting Gaussian process priors for complex computer codes. *Journal of the Royal Statistical Society Series D: The Statistician*, 51(1):81–97, 2002.

[115] L. S. Bastos and A. O'Hagan. Diagnostics for Gaussian process emulators. *Technometrics*, 51(4):425–438, 2009.

[116] I. Andrianakis and P. G. Challenor. The effect of the nugget on Gaussian process emulators of computer models. *Computational Statistics & Data Analysis*, 56 (12):4215–4228, dec 2012.

[117] T. J. Rogers, G. Manson, K. Worden, and E. J. Cross. On the choice of optimisation scheme for Gaussian process hyperparameters in SHM problems. In *Proceedings of the The 11th International Workshop on Structural Health Monitoring*, 2017.

[118] Jun Sun, Bin Feng, and Wenbo Xu. Particle swarm optimization with particles having quantum behavior. In *Proceedings of the 2004 Congress on Evolutionary Computation*, pages 325–331.

[119] S. S. Garud, I. A. Karimi, and M. Kraft. Design of computer experiments: a review. *Computers and Chemical Engineering*, 2017.

[120] H. Dette and A. Pepelyshev. Generalized latin hypercube design for computer experiments. *Technometrics*, 52(4):421–429, nov 2010.

[121] M. Liefvendahl and R. Stocki. A study on algorithms for optimization of Latin hypercubes. *Journal of Statistical Planning and Inference*, 136(9):3231–3247, 2006.

[122] J. Quinonero-Candela and C. E. Rasmussen. A unifying view of sparse approximate Gaussian process regression. *Journal of Machine Learning Research*, 6: 1939–1959, 2005.

[123] E. Snelson and Z. Ghahramani. Local and global sparse Gaussian process approximations. In *11th International Conference on Artificial Intelligence and Statistics*, 2007.

[124] T. D. Bui, J. Yan, and R. E. Turner. A unifying framework for Gaussian process pseudo-point approximations using power expectation propagation. *Journal of Machine Learning Research*, 18(1):3649–3720, 2017.

[125] E. Snelson and Z. Ghahramani. Sparse Gaussian processes using pseudo-inputs. In *Advances in neural information processing systems*, 2005.

[126] M. Bauer, M. van der Wilk, and C. E. Rasmussen. Understanding probabilistic sparse Gaussian process approximations. In *Advances in neural information processing systems*, pages 1533–1541, 2016.

[127] M. Titsias. Variational learning of inducing variables in sparse Gaussian processes. In *International Conference on Artificial Intelligence and Statistics*, 2009.

[128] J. Hensman, N. Fusi, and N. D. Lawrence. Gaussian processes for big data. In *Conference on Uncertainty in Artificial Intelligence*, pages 282–290, 2013.

[129] P. Hennig and C. J. Schuler. Entropy search for information-efficient global pptimization. *Journal of Machine Learning Research*, 13:1809–1837, 2012.

[130] J. Snoek, H. Larochelle, and R. P. Adams. Practical Bayesian optimization of machine learning algorithms. In *Advances in neural information processing systems*, pages 2951–2959, 2012.

[131] P. Gardner, C. Lord, and R. J. Barthorpe. Bayesian history matching for forward model-driven structural health monitoring. In *Proceedings of IMAC XXXVI*, 2018.

[132] P. D. Arendt, D. W. Apley, W. Chen, D. Lamb, and D. Gorsich. Improving identifiability in model calibration using multiple responses. *Journal of Mechanical Design*, 134(10), 2012.

[133] P. Boyle and M. Frean. Dependent Gaussian Processes. *Advances in Neural Information Processing Systems*, pages 217–224, 2005.

[134] P. Boyle and M. Frean. Multiple output Gaussian process regression. Technical report, 2005.

[135] M. Alvarez and N. D. Lawrence. Sparse convolved Gaussian processes for multi-output regression. In *Advances in neural information processing systems*, pages 57–64, 2009.

[136] A. C. Damianou and N. D. Lawrence. Deep Gaussian Processes. In *International Conference on Artificial Intelligence and Statistics*, 2012.

[137] S. K. Sachdeva, P. B. Nair, and A. J. Keane. Comparative study of projection schemes for stochastic finite element analysis. *Computer Methods in Applied Mechanics and Engineering*, 2006.

[138] M. Lazaro-Gredilla, M. Titsias, L. Miguel, M. Lázaro-Gredilla, M. Titsias, M. Lazaro-Gredilla, and M. Titsias. Variational heteroscedastic Gaussian process regression. In *Proceedings of International Conference on Machine Learning*, 2011.

[139] R. Frigola-Alcalde. *Bayesian Time Series Learning with Gaussian Processes*. PhD thesis, 2015.

[140] A. Girard, C. E. Rasmussen, J. Q. Candela, and R. Murray-Smith. Gaussian process priors with uncertain inputs-application to multiple-step ahead time series forecasting. In *A*, pages 545–552, 2003.

[141] R. Frigola and C. E. Rasmussen. Integrated pre-processing for bayesian nonlinear system identification with gaussian processes. *Proceedings of the IEEE Conference on Decision and Control*, (3):5371–5376, 2013.

[142] K. Worden, W. E. Becker, T. J. Rogers, and E. J. Cross. On the confidence bounds of Gaussian process NARX models and their higher-order frequency response functions. *Mechanical Systems and Signal Processing*, 104:188–223, 2018.

[143] S. Särkkä and J. Hartikainen. Infinite-dimensional Kalman filtering approach to spatio-temporal Gaussian process regression. *Proceedings of the 15th International Conference on Artificial Intelligence and Statistics*, pages 993–1006, 2012.

[144] A. Svensson, A. Solin, S. Särkkä, and T. B. Schön. Computationally efficient Bayesian learning of Gaussian process state space models. In *International Conference on Artificial Intelligence and Statistics*, 2015.

[145] P. D. Arendt, D. W. Apley, and W. Chen. A preposterior analysis to predict identifiability in the experimental calibration of computer models. *IIE Transactions*, 48(1):75–88, 2016.

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

[147] Q. Liu and D. A. Pierce. A note on Gauss-Hermite quadrature. *Biometrika*, 81(3):624–629, 1994.

[148] G. H. Golub and J. H. Welsch. Calculation of Gauss quadrature rules. *Mathematics of Computation*, 23(106):221–221, 1969.

[149] A. Gelman, J. B. Carlin, H. S. Stern, and D. B. Rubin. *Bayesian Data Analysis*. 2004.

[150] R. C. Smith. *Uncertainty quantification : theory, implementation, and applications*. Socciety for Industrial and Applied Mathematics, 2013.

[151] G. O. Roberts, A. Gelman, and W. R. Gilks. Weak convergence and optimal scaling of random walk Metropolis algorithms. Technical Report 1, 1997.

[152] H. Haario, M. Laine, A. Mira, and E. Saksman. DRAM: Efficient adaptive MCMC. *Statistics and Computing*, 16(4):339–354, 2006.

[153] D. S. Oliver and Y. Chen. Recent progress on reservoir history matching: a review. *Computational Geosciences*, 15(1):185–221, 2011.

[154] I. Vernon, M. Goldstein, and R. G. Bower. Galaxy formation: a Bayesian uncertainty analysis. *Bayesian Analysis*, 5(4):619–669, 2010.

[155] I. Vernon, M. Goldstein, and R. Bower. Galaxy formation: Bayesian history matching for the observable universe. *Statistical Science*, 29(1):81–90, feb 2014.

[156] N. R. Edwards, D. Cameron, and J. Rougier. Precalibrating an intermediate complexity climate model. *Climate Dynamics*, 37(7-8):1469–1482, 2011.

[157] D. Williamson, M. Goldstein, L. Allison, A. Blaker, P. Challenor, L. Jackson, and K. Yamazaki. History matching for exploring and reducing climate model parameter space using observations and a large perturbed physics ensemble. *Climate Dynamics*, 41(7-8):1703–1729, 2013.

[158] K. Worden, G. Manson, and N. R. J. Fieller. Damage detection using outlier analysis. *Journal of Sound and Vibration*, 229(3):647–667, 2000.

[159] F. Pukelsheim. The three sigma rule. *American Statistician*, 48(2):88–91, 1994.

[160] D. Williamson and I. Vernon. Efficient uniform designs for multi-wave computer experiments. pages 1–31, 2013.

[161] R. D. Wilkinson. Approximate Bayesian computation (ABC) gives exact results under the assumption of model error. *Statistical applications in genetics and molecular biology*, 12(2):129–41, 2013.

[162] D. R. Jones, M. Schonlau, and W. J. Welch. Efficient global optimization of expensive black-box functions. *Journal of Global Optimization*, 13(4):455–492, 1998.

[163] C. Chevalier, D. Ginsbourger, J. Bect, E. Vazquez, V. Picheny, and Y. Richet. Fast parallel kriging-based stepwise uncertainty reduction with application to the identification of an excursion set. *Technometrics*, 56(4):455–465, 2014.

[164] J. M. Hernández-Lobato, M. W. Hoffman, and Z. Ghahramani. Predictive entropy search for efficient global optimization of black-box functions. In *Advances in neural information processing systems*, pages 1–12, 2014.

[165] D. Petelin, G. Matej, and V. Smidl. Adaptive importance sampling for Bayesian inference in Gaussian process models. *IFAC Proceedings*, 47(3):5011–5016, 2014.

[166] A. Svensson, J. Dahlin, and T. B. Schön. Marginalizing Gaussian process hyperparameters using sequential Monte Carlo. In *IEEE 6th International Workshop on Computational Advances in Multi-Sensor Adaptive Processing*, 2015.

[167] A. E. Raftery, T. Gneiting, F. Balabdaoui, and M. Polakowski. Using Bayesian model averaging to calibrate forecast ensembles. *Monthly Weather Review*, 133 (5):1155–1174, 2005.

[168] Q. Duan, N. K. Ajami, X. Gao, and S. Sorooshian. Multi-model ensemble hydrologic prediction using Bayesian model averaging. *Advances in Water Resources*, 30(5):1371–1386, 2007.

[169] L. Bao, T. Gneiting, E. P. Grimit, P. Guttorp, and A. E. Raftery. Bias correction and Bayesian model averaging for ensemble forecasts of surface wind direction. *Monthly Weather Review*, 138(5):1811–1821, 2010.

[170] J. M. Montgomery, F. M. Hollenbach, and M. D. Ward. Improving predictions using ensemble Bayesian model averaging. *Political Analysis*, 20(3):271–291, 2012.

[171] B. Peherstorfer, K. Willcox, and M. Gunzburger. Survey of multifidelity methods in uncertainty propagation, inference, and optimization. *SIAM Review*, 60(3):550–591, 2018.

[172] J. B. Nagel and B. Sudret. A unified framework for multilevel uncertainty quantification in Bayesian inverse problems. *Probabilistic Engineering Mechanics*, 43:68–84, 2016.

[173] M. Strong, J. E. Oakley, and J. Chilcott. Managing structural uncertainty in health economic decision models. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 61(1):25–45, 2012.

[174] S. Christides and A. D. S. Barr. One-dimensional theory of cracked Bernoulli-Euler beams. *International Journal of Mechanical Sciences*, 26(11-12):639–648, 1984.

[175] J. Sinha, M. Friswell, and S. Edwards. Simplified models for the location of cracks in beam structures using measured vibration data. *Journal of Sound and Vibration*, 251(1):13–38, 2002.

[176] M. D. Todd, J. M. Nichols, C. J. Nichols, and L. N. Virgin. An assessment of modal property effectiveness in detecting bolted joint degradation: Theory and experiment. *Journal of Sound and Vibration*, 275(3-5):1113–1126, 2004.

[177] J. M. Nichols and K. D. Murphy. *Modeling and Estimation of Structural Damage*. John Wiley & Sons, Ltd, Chichester, UK, 2016.

[178] P. Gardner, R. J. Barthorpe, and C. Lord. The Development of a Damage Model for the use in Machine Learning Driven SHM and Comparison with Conventional SHM Methods. In *Proceedings of the International Conference on Noise and Vibration Engineering*, pages 3333–3346, 2016.